

UNIVERSIDADE FEDERAL DO PARANÁ

ADRIANO VITOR

UMA PROPOSTA DE ALGORITMO GENÉTICO HÍBRIDO PARA O  
PROBLEMA DO CAIXEIRO VIAJANTE

CURITIBA  
2015

ADRIANO VITOR

UMA PROPOSTA DE ALGORITMO GENÉTICO HÍBRIDO PARA O  
PROBLEMA DO CAIXEIRO VIAJANTE

Projeto de tese apresentado ao Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, linha de pesquisa em Abordagem de Problemas da Pesquisa Operacional, do Departamento de Matemática, Setor de Ciências Exatas e do Departamento de Construção Civil, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Doutor em Ciências.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Luzia Vidal de Souza

Coorientador: Prof. Dr. Paulo Henrique Siqueira

CURITIBA

2015

---

V845p

Vitor, Adriano

Uma proposta de algoritmo genético híbrido para o problema do caixeiro viajante/ Adriano Vitor. – Curitiba, 2015.

101f. : il. [algumas color.] ; 30 cm.

Tese (doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Setor de Tecnologia, Programa de Pós-graduação em Métodos Numéricos em Engenharia.

Orientadora: Luzia Vidal de Souza -- Coorientador: Paulo Henrique Siqueira.

1.Algoritmos genéticos. 2. Problema do caixeiro viajante. 3. Heurística 3-OPT. I. Souza, Luzia Vidal. II. Siqueira, Paulo Henrique. III. Título.

CDD: 518.1

---

## TERMO DE APROVAÇÃO

ADRIANO VITOR

### UMA PROPOSTA DE ALGORITMO GENÉTICO HÍBRIDO PARA O PROBLEMA DO CAIXEIRO VIAJANTE

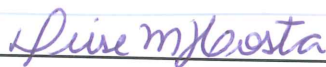
Tese aprovada como requisito parcial para obtenção do grau de doutor no Programa de Pós-Graduação em Métodos Numéricos em Engenharia, da Universidade Federal do Paraná, pela seguinte banca examinadora:



**Prof.ª Dr.ª Luzia Vidal de Souza**  
Orientadora – membro do PPGMNE/UFPR



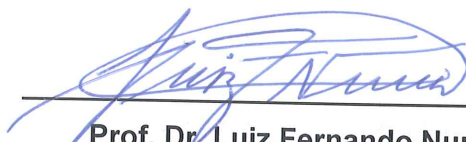
**Prof.ª Dr.ª Angela Olandoski Barboza**  
Membro do Dep. Acadêmico de Matemática da UTFPR



**Prof. Dr. Deise Maria Bertholdi Costa**  
Membro do PPGMNE/UFPR



**Prof. Dr. Leandro Magatão**  
Membro do Dep. Acadêmico de Mecânica da UTFPR



**Prof. Dr. Luiz Fernando Nunes**  
Membro do Dep. Acadêmico de Matemática da UTFPR

Curitiba, 30 de setembro de 2015.

Dedico este trabalho a toda minha família que me apoiou e soube compreender as abdições em prol do estudo. Em especial, à minha esposa Aline, pelo companheirismo e cumplicidade. Aos meus amigos Gislaine Perçaro, Juliano Motta, Sandro Rodrigues, Solange Regina e Tatiane Cazarin, por todos os momentos juntos, as discussões produtivas, as viagens e a parceria nos estudos.

## **AGRADECIMENTOS**

Agradeço a Deus porque sem sua permissão nenhum passo em minha vida seria dado.

Agradeço também a minha esposa Aline, que nos últimos 4 anos tem estado ao meu lado, paciente, atenciosa, amorosa e, muitas vezes, abdicando de nossa vida e de suas metas para que eu pudesse estudar e estar ausente do seio de nosso lar.

Não posso deixar de agradecer minha querida mãe, que apesar de seu pouco estudo e as dificuldades da vida, principalmente financeiras, sempre me incentivou a estudar, mas mais que isso, moldou meu caráter e contribuiu para fazer de mim o homem que sou hoje.

Aos meus amigos Camila Melo, Diogo Heron, Edivaldo Bottega, Gislaine Pericaro, Juliano Fabiano, Marina, Paulo Weinhardt, Rômulo, Sandro Rodrigues, Solange Regina, Tatiane Cazarin, Vitorio Kurpel e tantos outros, que compartilharam comigo diversos momentos desta caminhada e contribuíram imensamente durante toda a trajetória.

Ao professor Anselmo Chaves Neto, por seus ensinamentos, pelo apoio e incentivo a mim e a meus amigos durante nossas viagens a Curitiba para estudo.

Agradeço especialmente à minha orientadora Luzia Vidal de Souza e meu coorientador Paulo Henrique Siqueira que foram pacientes, souberam me entender quanto aos escorregões e percalços durante toda a caminhada.

Por fim agradeço ao IFSC por me conceder afastamento e permitir o término do doutorado com mais tranquilidade e dedicação, e ainda a UFPR por proporcionar as condições de estudo.

O que nos faz melhores não é necessariamente o intelecto, tão pouco os recursos, mas a capacidade de nos tornarmos cada vez mais humanos.

## RESUMO

Este trabalho apresenta a implementação de um Algoritmo Genético (AG) adaptado e um AG híbrido com a heurística 3-Opt para resolver o Problema do Caixeiro Viajante (PCV). Modificações nas regras internas de funcionamento de um AG clássico foram realizadas, transformando-o na versão denominada AGA (AG Adaptado) que posteriormente foi hibridizada por meio da versão denominada AGAH (AG Adaptado e Híbrido Com 3-Opt). Também são propostos três novos operadores genéticos de cruzamento (Cortes Não Paralelos com 4 Trechos (CNP4), Máxima Preservação Heurística (MPH) e Máxima Preservação Heurística - Versão b (MPHb)). O operador CNP4 foi associado ao AGA e ao AGAH enquanto que o MPH e o MPHb foram implementados somente para o AGAH com a finalidade de contribuir com o ajuste fino das soluções. A heurística 3-Opt também foi incorporada ao AGAH, o que lhe conferiu o caráter híbrido. Esta heurística foi modificada para considerar a ordem com que as cidades são escolhidas ao realizar trocas de arestas inerentes a aplicação da mesma. O critério apresentado determina a ordem das cidades, ranqueando-as por duas medidas estatísticas: a distância média de cada cidade aos seus vizinhos mais próximos e o desvio padrão obtido do mesmo conjunto de vizinhos. Para validar a eficácia das modificações, propôs-se 4 versões de AGs (AG Trechos, AG Rinap, AG Acs, AG Padronizar) contendo as modificações individualmente. O desempenho destas versões foi comparado ao desempenho de um AG clássico e, na sequência, elas foram agrupadas para formar o AGA, cujo desempenho também foi avaliado. Para validar a eficiência do AGAH foram realizados testes computacionais com a biblioteca de problemas disponível na internet e conhecida como *Traveling Salesman Problem Library* (TSPLIB). Os resultados encontrados foram comparados aos obtidos pelo algoritmo LK-H de Helsgaun (2000) que representa uma das mais poderosas heurísticas para resolução do PCV. O AGAH se mostrou muito eficaz para instâncias com até 400 cidades, chegando a encontrar a solução ótima em 100% das execuções para instâncias em que o LK-H não apresentou o mesmo sucesso. A qualidade das soluções para problemas maiores foi inferior às obtidas pelo LK-H, principalmente devido a restrições de tempo de processamento. Ao final são recomendadas ações para trabalhos futuros que podem tanto melhorar o tempo de execução quanto a qualidade das soluções para instâncias de grande porte, também são sugeridos testes para avaliar se as adaptações presentes no AGA agregam melhor desempenho aos AGs em outras aplicações.

Palavras-chave: Otimização Combinatória. Programação Matemática. Métodos Heurísticos. Problema do Caixeiro Viajante.



## ABSTRACT

This work presents the implementation of a adapted Genetic Algorithm (GA) and one GA hybrid with the 3-Opt heuristic to solve the Traveling Salesman Problem (TSP). Changes in the rules of procedure of a classic GA were made, turning it into the version denominated as AGA (Adapted GA) which was subsequently hybridized generating AGAH (adapted GA hybrid with 3-Opt). It is also proposed three new crossing genetic operators (Non Parallel Cuts with 4 Stretches (CNP4), Maximum Heuristic Preservation (MPH) and Maximum Heuristic Preservation version b (MPH-b)). The operator CNP4 was associated with AGA and AGAH while MPH and MPH-b were implemented only to AGAH for contributing to the fine adjustment of the solutions. The 3-Opt heuristic was also incorporated to AGAH, which confers it the hybrid character. This heuristic was modified to consider the order in which the cities are selected while performing the edges exchanges inherent to the application of itself. The criteria presented determines the order of the cities, ranking them by two statistical measures: the average distance of each city to its closest neighbors and the standard deviation obtained from the same set of neighbors. In order to validate the efficiency of the modifications, it was proposed four version of GAs (GA Stretches, GA Rinap, GA Acs, GA Standardize) containing the modifications individually. The performance of these versions were compared to that of a classic GA and, subsequently, they were clustered to form AGA, whose performance was also evaluated. In order to validate the efficiency of AGAH is was conducted computational tests with the problem library available on the internet known as Traveling Salesman Problem Library (TSPLIB). The results obtained were compared to the ones acquired by the algorithm LK-H from Helsgaun (2000), which represents one of the most powerful heuristics for the resolution of the TSP. AGAH has proven to be very effective for instances with up to 400 cities, being able to find the optimal solution in 100% of the executions for instances in which LK H did not present the same result. The quality of solutions for larger problems was inferior to those obtained by LK-H, mostly due to restriction in processing time. In the end, actions are recommended for future works, which can improve both execution time and quality of solutions for large instances. Tests are also suggested to evaluate if the adaptations present in AGA aggregate better performance to the GAs in other applications.

Key-words: Combinatorial Optimization. Mathematical Programming. Heuristic Methods. Traveling Salesman Problem.

## LISTA DE FIGURAS

FIGURA 1 – EXEMPLO DE PCV COM 48 CIDADES .....	31
FIGURA 2 – EXEMPLO DE PLACAS DE CIRCUITO IMPRESSO .....	32
FIGURA 3 – PASSO DE 3-OPT SOBRE UM PCV .....	38
FIGURA 4 – PASSOS DE UM AG CLÁSSICO .....	40
FIGURA 5 – REPRESENTAÇÃO POR CAMINHO PARA UM PCV .....	41
FIGURA 6 – CONFIGURAÇÕES DIFERENTES PARA UMA MESMA SOLUÇÃO	41
FIGURA 7 – ETAPAS 1 E 2 DA GERAÇÃO DA SOLUÇÃO .....	45
FIGURA 8 – ETAPAS 3 E 4 DA GERAÇÃO DA SOLUÇÃO .....	45
FIGURA 9 – ETAPAS 5 E 6 DA GERAÇÃO DA SOLUÇÃO .....	46
FIGURA 10 – ETAPAS 7 E 8 DA GERAÇÃO DA SOLUÇÃO .....	46
FIGURA 11 – ETAPAS 9 E 10 DA GERAÇÃO DA SOLUÇÃO .....	47
FIGURA 12 – CÁLCULO DA SIMILARIDADE ENTRE DUAS SOLUÇÕES .....	49
FIGURA 13 – CONSTRUÇÃO DE DESCENDENTES COM OPERADOR PMX ..	51
FIGURA 14 – CONSTRUÇÃO DE DESCENDENTES COM OPERADOR OX ....	52
FIGURA 15 – MOVIMENTOS BÁSICOS ENTRE OS PAIS .....	53
FIGURA 16 – CONSTRUÇÃO DOS CICLOS PARA AUXILIAR O OPERADOR CX	53
FIGURA 17 – CONSTRUÇÃO DE DESCENDENTES COM O OPERADOR CX .	54
FIGURA 18 – CONSTRUÇÃO DE DESCENDENTES COM O OPERADOR MPX	54
FIGURA 19 – CONSTRUÇÃO DA LISTA PARA APLICAR O OPERADOR ERX .	55
FIGURA 20 – RESULTADO DO PASSO 1 DO OPERADOR CNP4 .....	56
FIGURA 21 – RESULTADO DO PASSO 2 DO OPERADOR CNP4 .....	57
FIGURA 22 – RESULTADO DO PASSO 3 DO OPERADOR CNP4 .....	57
FIGURA 23 – RESULTADO DO PASSO 4 DO OPERADOR CNP4 .....	57

FIGURA 24 – RESULTADO DO PASSO 5 DO OPERADOR CNP4 .....	58
FIGURA 25 – RESULTADO DO PASSO 1 DO OPERADOR MPH .....	59
FIGURA 26 – SUBPROBLEMA DO PASSO 2 DO OPERADOR MPH .....	60
FIGURA 27 – SOLUÇÃO PARA O SUBPROBLEMA APÓS 3-OPT .....	60
FIGURA 28 – FILHO GERADO PELO OPERADOR MPH .....	61
FIGURA 29 – EXEMPLO DE MUTAÇÃO POR DESLOCAMENTO .....	62
FIGURA 30 – EXEMPLO DE MUTAÇÃO POR TROCA .....	63
FIGURA 31 – EXEMPLO DE MUTAÇÃO POR INSERÇÃO .....	63
FIGURA 32 – EXEMPLO DE MUTAÇÃO POR SIMPLES INVERSÃO .....	64
FIGURA 33 – EXEMPLO DE MUTAÇÃO POR INVERSÃO .....	64
FIGURA 34 – FLUXOGRAMA DO AGA .....	67
FIGURA 35 – PADRONIZAÇÃO DA POPULAÇÃO .....	71
FIGURA 36 – FLUXOGRAMA DO AGAH .....	73
FIGURA 37 – EXEMPLO PARA APLICAR O CRITÉRIO ESTATÍSTICO .....	78
FIGURA 38 – RESULTADO DA ANÁLISE ESTATÍSTICA ENTRE O AG TRECHOS E O AG CLÁSSICO .....	81
FIGURA 39 – TEMPO DE EXECUÇÃO DO AGATRECHOS E DO AGCLÁSSICO	81
FIGURA 40 – RESULTADO DA ANÁLISE ESTATÍSTICA ENTRE O AG ACS E O AG CLÁSSICO .....	82
FIGURA 41 – TEMPO DE EXECUÇÃO DO AG ACS E DO AG CLÁSSICO .....	82
FIGURA 42 – RESULTADO DA ANÁLISE ESTATÍSTICA ENTRE O AG RINAP E O AG CLÁSSICO .....	83
FIGURA 43 – TEMPO DE EXECUÇÃO DO AG RINAP E DO AG CLÁSSICO ...	83
FIGURA 44 – RESULTADO DA ANÁLISE ESTATÍSTICA ENTRE O AG PADRONI- ZAR E O AG CLÁSSICO .....	84
FIGURA 45 – TEMPO DE EXECUÇÃO DO AG PADRONIZAR E O AG CLÁS- SICO .....	84
FIGURA 46 – RESULTADO DA ANÁLISE ESTATÍSTICA ENTRE O AG E O AGA	87

FIGURA 47 – TEMPO DE EXECUÇÃO DO AG CLÁSSICO E DO AGA .....	88
FIGURA 48 – TEMPO DE EXECUÇÃO DA HEURÍSTICA 3-OPT .....	90

## LISTA DE TABELAS

TABELA 1	– EVOLUÇÃO NA RESOLUÇÃO DO PCV - MÉTODOS EXATOS ..	20
TABELA 2	– TERMOS FREQUENTES EM HEURÍSTICAS .....	36
TABELA 3	– TERMOS UTILIZADOS EM AGS E NA BIOLOGIA .....	39
TABELA 4	– CONSTRUÇÃO DA REPRESENTAÇÃO ORDINAL .....	42
TABELA 5	– REPRESENTAÇÃO DE UMA SOLUÇÃO POR MATRIZ ESPARSA	42
TABELA 6	– PRINCIPAIS OPERADORES GENÉTICOS ATÉ 1999 .....	50
TABELA 7	– PARÂMETROS PARA O AGA PROPOSTO NESTE TRABALHO	68
TABELA 8	– INSTÂNCIAS DO TSPLIB NÃO RESOLVIDAS PELO AGAH .....	74
TABELA 9	– PARÂMETROS PARA O AGAH PROPOSTO NESTE TRABALHO	75
TABELA 10	– DISTÂNCIAS ENTRE AS CIDADES DA FIGURA 37 .....	78
TABELA 11	– COMPARAÇÃO ENTRE AG CLÁSSICO E AGA .....	85
TABELA 12	– TESTE ESTATÍSTICO AG CLÁSSICO × AGA .....	86
TABELA 13	– 3-OPT COM CRITÉRIO × 3-OPT SEM CRITÉRIO .....	88
TABELA 14	– AGAH × LK-H - PROBLEMAS COM ATÉ 400 CIDADES .....	91
TABELA 15	– AGAH × LK-H - PROBLEMAS COM MAIS DE 400 CIDADES ...	92

## LISTA DE SIGLAS

AG	Algoritmo Genético
PCV	Problema do Caixeiro Viajante
AGA	AG Adaptado
AGAH	AG Adaptado e Híbrido Com 3-Opt
CNP4	Cortes Não Paralelos com 4 Trechos
MPH	Máxima Preservação Heurística
MPHb	Máxima Preservação Heurística - Versão b
TSPLIB	<i>Traveling Salesman Problem Library</i>
LK-H	Algoritmo de Lin e Kernighan - Heslgaun
PSO	<i>Particle Swarm Optimization</i>
ACO	<i>Ant Colony Optimization</i>
GPU	<i>Graphics Processing Unit</i>
CPU	<i>Central Processing Unit</i>
PCI	Placas de Circuito Impresso
LK	Algoritmo de Lin e Kernighan
LK-ABCC	Algoritmo de Lin e Kernighan - Applegate, Bixby, Chvátal e Cook
LK-NYYY	Algoritmo de Lin e Kernighan - Nguyen, Yoshihara, Yamamori e Yasunaga
LK-N	Algoritmo de Lin e Kernighan - Neto
1PX	1 <i>Crossover Point</i>
ACS	Aplicar Critério de Similaridade
SIM	<i>Simple Inversion Mutation</i>
PMX	<i>Partially - Mapped Crossover</i>
OX	<i>Order - Crossover</i>
CX	<i>Cycle Crossover</i>
HX	<i>Heuristic Crossover</i>
MPX	<i>Maximal Preservative Crossover</i>
ISM	<i>Insertion Mutation</i>
ERX	<i>Edge Recombination Crossover</i>

VR	<i>Voting - Recombination Crossover</i>
EM	<i>Exchange Mutation</i>
IVM	<i>Inversion Mutation</i>
SM	<i>Seramble Mutation</i>
OBX	<i>Order Based Crossover</i>
POS	<i>Position Based Crossover</i>
DM	<i>Displacement Mutation</i>
AP	<i>Alternating - Position Crossover</i>
RINAP	Reinserção de Indivíduos Novos e Aleatórios na População
MenEx	Menor Excesso
MedEx	Excesso Médio
MaxEx	Maior Excesso

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>18</b>
1.1 OBJETIVOS .....	18
1.1.1 Objetivo Geral .....	18
1.1.2 Objetivos Específicos .....	19
1.2 JUSTIFICATIVA .....	19
1.3 IMPORTÂNCIA DO TRABALHO .....	20
1.4 LIMITAÇÕES DO TRABALHO .....	21
1.5 INOVAÇÕES PROPOSTAS .....	21
1.6 ESTRUTURA DO TRABALHO .....	22
<b>2 REVISÃO DE LITERATURA .....</b>	<b>23</b>
2.1 METODOLOGIAS EXATAS PARA O PCV .....	23
2.2 METODOLOGIAS HEURÍSTICAS PARA O PCV .....	24
2.3 CONTRIBUIÇÕES EM ALGORITMOS GENÉTICOS .....	26
2.3.1 Resultados Teóricos e o Teorema do Esquema .....	27
2.4 PROCESSAMENTO EM PARALELO E MÉTODOS HÍBRIDOS .....	28
<b>3 MATERIAL E MÉTODO .....</b>	<b>30</b>
3.1 O PROBLEMA DO CAIXEIRO VIAJANTE .....	30
3.2 APLICAÇÕES REAIS DO PCV .....	31
3.2.1 Perfuração de Placas de Circuito Impresso .....	31
3.2.2 Análise de Estruturas de Cristais .....	32
3.2.3 Manutenção de Sistemas de Motores de Turbina a Gás .....	32
3.2.4 Manipulação de Materiais em um Armazém .....	33
3.3 MÉTODOS EXATOS PARA RESOLUÇÃO DO PCV .....	33



3.3.1	Formulação exata do PCV .....	34
3.3.2	Método <i>Branch-and-Bound</i> .....	35
3.3.3	Método <i>Branch-and-Cut</i> .....	35
3.4	MÉTODOS HEURÍSTICOS PARA RESOLUÇÃO DO PCV .....	36
3.4.1	Vizinho Mais Próximo .....	37
3.4.2	Heurística de Melhoria k-Opt .....	37
3.4.3	Algoritmo de Lin e Kernighan e suas Variações .....	38
3.5	ALGORITMO GENÉTICO .....	39
3.5.1	Características Gerais do Método .....	39
3.5.2	Representação das Soluções no Espaço de Busca .....	41
3.5.2.1	Representação por Caminho .....	41
3.5.2.2	Representação Ordinal .....	41
3.5.2.3	Representação por Matriz Esparsa .....	42
3.5.3	Função de Aptidão .....	43
3.5.4	População Inicial .....	43
3.5.4.1	Método do Vizinho Mais Próximo Adaptado .....	44
3.5.5	Métodos de Seleção .....	47
3.5.6	Operadores Genéticos - Cruzamento .....	50
3.5.6.1	Operador PMX .....	51
3.5.6.2	Operador OX .....	52
3.5.6.3	Operador CX .....	53
3.5.6.4	Operador MPX .....	54
3.5.6.5	Operador ERX .....	54
3.5.6.6	Operador CNP4 .....	55
3.5.6.7	Operador MPH .....	58
3.5.6.8	Operador MPHb .....	61

3.5.7	Operadores Genéticos - Mutação .....	62
3.5.7.1	Mutação por Deslocamento .....	62
3.5.7.2	Mutação por Troca .....	63
3.5.7.3	Mutação por Inserção .....	63
3.5.7.4	Mutação por Simples Inversão .....	63
3.5.7.5	Mutação por Inversão .....	64
3.5.8	Elitismo .....	64
3.6	CONCLUSÕES DO CAPÍTULO .....	65
<b>4</b>	<b>IMPLEMENTAÇÃO COMPUTACIONAL .....</b>	<b>66</b>
4.1	DESCRIÇÃO DO AGA PROPOSTO E SUAS VARIAÇÕES .....	66
4.1.1	Banco de Dados para os Testes .....	67
4.1.2	Informações de <i>Hardware</i> e <i>Software</i> .....	68
4.1.3	Parâmetros do AGA .....	68
4.1.4	Estrutura de Dados Adotada .....	69
4.1.5	Geração da População Inicial .....	69
4.1.6	Função de Aptidão da População .....	69
4.1.7	Rotina RINAP .....	69
4.1.8	Método de Seleção para Cruzamento e Mutação .....	70
4.1.9	Operadores de Cruzamento .....	70
4.1.10	Operadores de Mutação .....	70
4.1.11	Critério para Avaliar os Sobreviventes .....	70
4.1.12	Critério de Padronização da População .....	71
4.2	DESCRIÇÃO DO AGAH PROPOSTO .....	71
4.2.1	Banco de Dados para os Testes .....	74
4.2.2	Informações de <i>Hardware</i> e <i>Software</i> .....	74
4.2.3	Parâmetros do AGAH .....	74

4.2.4	Estrutura de Dados Adotada .....	75
4.2.5	Geração da População Inicial .....	76
4.2.6	Função de Aptidão da População .....	76
4.2.7	Rotina RINAP .....	76
4.2.8	Método de Seleção para Cruzamento e Mutação .....	76
4.2.9	Operadores de Cruzamento .....	76
4.2.10	Operadores de Mutação .....	76
4.2.11	Heurística 3-Opt .....	77
4.2.12	Critério de Padronização da População .....	79
<b>5</b>	<b>DISCUSSÕES E RESULTADOS COMPUTACIONAIS .....</b>	<b>80</b>
5.1	ANÁLISE DAS MODIFICAÇÕES PROPOSTAS SOBRE O AG CLÁSSICO ..	80
5.1.1	Comparação Entre AG Trechos e o AG Clássico .....	80
5.1.2	Comparação Entre AG Acs e o AG Clássico .....	82
5.1.3	Comparação Entre AG Rinap e o AG Clássico .....	83
5.1.4	Comparação Entre AG Padronizar e o AG Clássico .....	84
5.2	COMPARAÇÃO ENTRE AGA E O AG CLÁSSICO .....	85
5.3	MODIFICAÇÕES PROPOSTAS NA UTILIZAÇÃO DA HEURÍSTICA 3-OPT .	88
5.4	COMPARAÇÃO ENTRE O AGAH E O LK-H .....	91
<b>6</b>	<b>CONCLUSÕES E RECOMENDAÇÕES .....</b>	<b>94</b>
6.1	CONSIDERAÇÕES SOBRE O TRABALHO DESENVOLVIDO .....	94
6.2	RECOMENDAÇÕES PARA TRABALHOS FUTUROS .....	95
	<b>REFERÊNCIAS .....</b>	<b>97</b>

## 1 INTRODUÇÃO

O mundo moderno exige das pessoas, instituições e governos, respostas rápidas e precisas para seus problemas. Estes podem ser resolvidos a partir de métodos da Pesquisa Operacional que oferece suporte para a abordagem de uma série de situações por meio de suas técnicas efetivas para tratar problemas reais.

Neste contexto existem diversas aplicações envolvendo uma classe importante de problemas, denominados NP-Completo, que recebem este nome porque não se pode provar que existe um algoritmo capaz de resolver até a otimalidade instâncias de qualquer tamanho em tempo estimado por uma função polinomial (MICHAEL; DAVID, 1979).

No universo destes problemas, possivelmente o PCV é o mais estudado por ter grande importância prática e ser de fácil formulação. Em linhas gerais, o mesmo consiste em determinar o melhor roteiro a ser realizado para visitar um conjunto de cidades. No entanto, tal problema é de difícil resolução até a otimalidade, ou seja, à medida que mais cidades são adicionadas ao roteiro, a maioria das metodologias encontram dificuldade para determinar a solução ótima.

Segundo Applegate *et al.* (2007), a origem do nome PCV é incerta. Contudo, sua popularização ocorreu na década de 1950 e desde então diversas metodologias têm sido propostas para a resolução deste problema, capazes de apresentar boas soluções para instâncias cada vez maiores em tempos aceitáveis.

Dentre as metodologias existentes, será discutido neste trabalho, a heurística 3-Opt e os AGs, com o intuito de agrupá-las em um procedimento híbrido, aproveitando as potencialidades de ambas. Para promover maior efetividade, modificações tanto no AG quanto na heurística 3-Opt serão realizadas. Outras abordagens híbridas são relatadas em Nguyen *et al.* (2007b) e Kuroda *et al.* (2010).

### 1.1 OBJETIVOS

#### 1.1.1 OBJETIVO GERAL

Um dos objetivos principais deste trabalho é revisar as regras internas de funcionamento de um AG, promovendo ganho de desempenho desta metodologia na resolução de PCVs. Também objetiva-se propor uma metodologia híbrida, associando

um Algoritmo Genético adaptado à heurística 3-Opt, que apresente resultados efetivos para resolução de instâncias com até 400 cidades.

### 1.1.2 OBJETIVOS ESPECÍFICOS

- 1) Revisar as regras internas de funcionamento de um AG, com o propósito de torná-lo mais eficaz na busca por soluções para PCVs.
- 2) Propor três novos operadores genéticos de cruzamento para o PCV (CNP4, MPH e MPHb).
- 3) Propor um novo método para determinar os indivíduos para cruzamento que busca a manutenção da diversidade genética e que não considere explicitamente o *fitness* dos indivíduos.
- 4) Propor um AG híbrido para resolver PCVs, que agrega as revisões nas regras internas de um AG clássico, novos operadores genéticos e a heurística 3-Opt de forma a se tornar competitivo em relação a outras heurísticas relatadas na literatura, como por exemplo o LK-H.
- 5) Elaborar um critério para a heurística 3-Opt proporcionando ganho de desempenho ao determinar a melhor ordem em que as cidades são selecionadas para se realizar as trocas.
- 6) Avaliar, por meio de testes numéricos e estatísticos, os benefícios individuais e o benefício coletivo agregados aos AGs, após as revisões de suas regras internas.

## 1.2 JUSTIFICATIVA

Com o constante desenvolvimento de processadores que os tornem cada vez mais velozes, a capacidade de resolução de instâncias consideráveis para o PCV com métodos exatos aumentou significativamente. Contudo, o tempo de processamento requerido ainda é muito alto. A Tabela 1 mostra a evolução histórica das instâncias resolvidas até a otimalidade utilizando métodos exatos. São apresentadas instâncias selecionadas de um conjunto com 111 problemas, presentes na TSPLIB, que foram propostas por diversos pesquisadores e reunidas na internet por Reinelt (1990).

TABELA 1: Evolução na Resolução do PCV - Métodos Exatos

Ano	Instância	Tamanho	Primeiros Pesquisadores a Obter a Solução Ótima
1954	dantzig42	42	Dantzig, Fulkerson e Johnson (1954)
1962	aleatória	64	Held e Karp (1962)
1974	aleatória	67	Camerine, Fratta e Maffioli (1974)
1980	gr120	120	Grötschel (1980)
1980	lin318	318	Crowder e Padberg (1980)
1987	att532	532	Padberg e Rinaldi (1987)
1991	gr666	666	Grötschel e Holland (1991)
1991	pr2392	2392	Padberg e Rinaldi (1991)
1995	pla7397	7397	Applegate, Bixby, Chvátal e Cook (1995)
1998	usa13509	13509	Applegate, Bixby, Chvátal e Cook (1998)
2001	d15112	15112	Applegate, Bixby, Chvátal e Cook (2001)
2004	sw24978	24978	Applegate, Bixby, Chvátal, Cook e Helsgaun (2004)

FONTE: Adaptado de Prestes (2006)

Para chegar a solução ótima da instância sw24978, por exemplo, foram necessários 3 meses de processamento em um *cluster* com 10 supercomputadores. Desta forma, justifica-se a implementação de novas heurísticas em busca de soluções adequadas com um tempo computacional aceitável.

Outra justificativa para a proposta do AGAH deste trabalho é que o mesmo apresenta uma metodologia atual e promissora. Atual porque as publicações mais recentes e notáveis da literatura têm abordado o mesmo tema, tais como em Wang (2014), Tsai *et al.* (2014), Honda, Nagata e Ono (2013), Liu e Zeng (2009) e Nguyen *et al.* (2007b) e promissora porque nestes trabalhos citados foram encontrados resultados adequados para instâncias de grande porte (acima de 1000 pontos) e, para algumas delas, os menores trajetos obtidos até então.

### 1.3 IMPORTÂNCIA DO TRABALHO

Este trabalho apresenta contribuições à comunidade científica por promover ganho de desempenho do AG puro, por meio da versão AGA que revisa suas regras de funcionamento, e por apresentar resultados satisfatórios com a com a versão híbrida AGAH.

Também é relevante porque propõe três novos operadores para o AG com bom desempenho diante dos operadores clássicos presentes na literatura, principalmente no ajuste fino das soluções.

Por fim, há relevância neste trabalho por apresentar indícios de ser mais robusto que heurísticas tais como o LK-H no sentido de encontrar a solução ótima com maior frequência para problemas com até 400 cidades.

## 1.4 LIMITAÇÕES DO TRABALHO

Uma das limitações deste trabalho é a forma de representação dos dados, já que para a heurística 3-Opt, a representação mais promissora é a estrutura de árvore binária por permitir avaliar trocas a um tempo reduzido. Para implementar um AG que resolve o PCV, é mais prático manipular as soluções usando a representação por caminho, o que conduz a uma escolha conflitante porque a estrutura mais adequada não é a mesma para ambas as metodologias implementadas neste trabalho.

Outra limitação relaciona-se à implementação da estrutura de árvore em um AG que resolve um PCV, pois se trata de um código complexo de ser construído (NGUYEN *et al.*, 2007a). Além do que, não existem operadores genéticos para esta estrutura e todo um estudo para criá-los deveria ser realizado para uma execução híbrida sem a necessidade de alternar entre a estrutura de árvore e a representação por caminho.

O tempo computacional requerido por implementações de AGs, constitui-se também numa limitação do trabalho, se comparados com os tempos requeridos por heurísticas tais como LK-H e similares, que são capazes de apresentar resultados adequados e requerem cerca de 10 vezes menos tempo que os AGs.

## 1.5 INOVAÇÕES PROPOSTAS

Em suas conclusões, Rave e Álvarez (2013) apontam evidências de duas tendências atuais: buscar avanços na área propondo metodologias híbridas que envolvem métodos existentes ou investir em novos campos de inspiração para dar origem a novos métodos, permitindo progressos na otimização combinatória. O AGAH aqui proposto segue a tendência de combinar métodos já existentes, melhorando seu desempenho, por alterar suas regras de funcionamento.

A primeira inovação proposta neste trabalho refere-se à implementação dos novos operadores CNP4, MPH e MPHb, objetivando contribuir para o ajuste fino das soluções, ou seja, possibilitar que após a população apresentar diversos indivíduos muito aptos, porém igualmente parecidos, ainda seja possível obter melhores indivíduos a partir deles.

Outra inovação proposta é a inserção de critérios estatísticos para determinar a ordem das cidades nas quais são realizadas as trocas com a heurística 3-Opt. Testes computacionais sugerem a relevância desta implementação para obtenção de melhores resultados em boa parte dos casos.

A utilização de um novo método de seleção de indivíduos para cruzamento, sem considerar explicitamente seus fitness, também representa uma inovação proposta.

Também foram implementadas adaptações nas regras de funcionamento do AG que representam inovações para tornar o método mais eficaz. Tais adaptações poderiam ser avaliadas em implementações de AGs para resolver outros problemas por não se relacionarem a particularidades do PCV.

## 1.6 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos. O primeiro capítulo contém: a introdução, os objetivos, as justificativas, a importância, as limitações, as inovações propostas e a presente estrutura do trabalho.

Com o segundo, é apresentada uma revisão de literatura percorrendo historicamente os avanços científicos mais relevantes relativos ao tema.

Por meio do capítulo três desenvolvem-se os materiais e métodos, detalhando a abordagem para o PCV proposta nesta tese por meio de descrições, algoritmos e exemplos. Também são apresentadas aplicações reais para motivar o estudo do problema.

Para o quarto capítulo expõe-se a implementação computacional necessária para atingir os objetivos geral e específicos propostos inicialmente.

No quinto capítulo são relatados e discutidos os resultados, as análises e comparações com outras metodologias.

No último capítulo são feitas as conclusões e recomendações para trabalhos futuros.



## 2 REVISÃO DE LITERATURA

### 2.1 METODOLOGIAS EXATAS PARA O PCV

A primeira publicação relatando a resolução de um PCV com 42 cidades foi proposta por Dantzig, Fulkerson e Johnson (1954). Os autores afirmam em seu trabalho que a origem do PCV é incerta mas que há vários anos o problema, não necessariamente com este nome, fazia parte de discussões em encontros da comunidade científica. Para resolver o problema, foi elaborada uma estratégia considerada uma das versões pioneiras do que passou a ser chamado de Método de Planos de Corte. Esta foi a primeira resolução até a otimalidade de um PCV por um método exato.

Ao longo dos anos, diversos autores obtiveram, por meio de métodos exatos, as soluções ótimas para instâncias cada vez maiores. Além do trabalho de Dantzig, Fulkerson e Johnson (1954) que, previamente mencionado, cronologicamente os seguintes trabalhos apresentam resultados semelhantes: Held e Karp (1962) para uma instância aleatória com 64 cidades, Grötschel (1980) para a instância gr120, Crowder e Padberg (1980) para instância lin318, Padberg e Rinaldi (1987) para a instância att532, Grötschel e Holland (1991) para a instância gr666, Padberg e Rinaldi (1991) para a instância pr2392, Applegate *et al.* (1995) para a instância pla7397, Applegate *et al.* (1998) para a instância usa13509, Applegate, Bixby e Chvátal (2001) para a instância d15112, Applegate *et al.* (2004) para a instância sw24978, todas encontradas na TSPLIB.

Por se tratar de um problema NP-Completo, conforme menciona Michael e David (1979), obter a solução ótima para grandes instância do PCV sempre foi um desafio para os pesquisadores. Felizmente, desde 1954 até os dias de hoje, muito se avançou em termos de metodologias exatas e heurísticas para abordar o problema. O Avanço de ferramentas ligadas a computação (aumento de memória RAM, computadores com vários núcleos de processadores possibilitando o processamento em paralelo), também contribuíram para se atingir os níveis atuais.

Em 1970, Held e Karp (1970) apresentaram uma relaxação no modelo exato para o PCV, transformando-o num problema de programação linear cuja solução não constitui uma solução viável, mas o valor da função objetivo representa um limite inferior para o valor ótimo do problema original. O problema linear obtido é polinomial e possível de ser resolvido. A partir de então, é comum os pesquisadores avaliarem se soluções encontradas em metodologias heurísticas estão próximas da solução ótima

para grandes instâncias do PCV por meio desse limite inferior.

Em 1990, Reinelt (1990) criou uma biblioteca contendo várias instâncias do PCV que vinham sendo (e são até hoje) testadas e discutidas na literatura. Essa biblioteca é conhecida como TSPLIB e contém mais de 100 exemplos com tamanhos que variam de 14 até 85.900 cidades.

## 2.2 METODOLOGIAS HEURÍSTICAS PARA O PCV

No trabalho de Lin e Kernighan (1973) foi apresentado um algoritmo heurístico denominado algoritmo LK, que futuramente foi revisado por diversos autores e, atualmente, algumas destas revisões estão entre as metodologias mais consolidadas para resolver grandes instâncias do PCV por consumir os menores tempos computacionais e apresentar soluções com menos de 0,5% de distância dos limites inferiores de Held e Karp (1970).

Em 1983 o *Simulated Annealing* foi proposto por Kirkpatrick, Gelatt e Vecchi (1983), inspirados em trabalhos anteriores da década de 1950 que simulavam o resfriamento de um material em um banho térmico (processo conhecido como *annealing* ou têmpera). Um tipo de tratamento térmico que consiste em submeter certos materiais inicialmente a altas temperaturas e reduzir gradualmente até atingir o equilíbrio térmico, com aumentos e reduções do estado de energia, tornando-os consistentes e rígidos.

No modelo computacional, o processo de diversificação acontece quando a temperatura está alta, fazendo com que o algoritmo explore um maior espaço de soluções. À medida que a temperatura baixa, a aceitação de soluções piores vai reduzindo, intensificando a busca, convergindo para soluções que reduzam a função de custo (PRESTES, 2006).

Glover (1989) propõe a heurística denominada Busca Tabu, sendo aplicada ao PCV em Knox e Glover (1989) e Knox (1994). O procedimento consiste em admitir uma solução inicial e a partir de movimentos na sua vizinhança, realizar uma pesquisa no espaço de soluções a procura da solução ótima. Um sistema de listas tabu impede que a busca se prenda a ótimos locais ou retorne a uma solução previamente visitada.

Para justificar o desenvolvimento de estratégias para resolver grandes instâncias do PCV é interessante que existam aplicações reais que requeiram este esforço. Em Grötschel, Jünger e Reinelt (1991) esta questão foi abordada ao apresentarem uma aplicação na construção de Placas de Circuito Impresso, cujo processo pode requerer a resolução de um PCV com algumas dezenas de milhares de pontos para

determinar a forma ótima de deslocar uma máquina sobre uma placa de circuito impresso para realizar furos necessários a afixação de componentes.

Jünger, Rinaldi e Reinelt (1995) apresentam diversas outras aplicações reais para o problema do caixeiro viajante, tais como: manutenção de turbinas a gás, problema de separação de pedidos em armazéns, programação da linha de produção, roteamento de veículos, Máscara de plotagem na produção de PCs e Controle de movimentos de robôs.

Johnson e McGeoch (1997) discorrem sobre diversas metodologias heurísticas para a resolução do PCV que incluem Algoritmos Genéticos, Busca Tabu, *Simulated Annealing* e as heurísticas de melhorias 2-Opt e 3-Opt.

A Otimização por Nuvem de Partículas, do inglês *Particle Swarm Optimization* (PSO), é uma abordagem baseada em população proposta inicialmente por Kennedy e Eberhart (1995) com base nos estudos de Reynolds (1987) e Heppner e Grenander (1990) que estudaram modelos de simulações de vôos de pássaros. Segundo Prestes (2006), um algoritmo PSO é iniciado com um grupo de partículas aleatórias e procura por soluções ótimas atualizando as partículas. Nas iterações, cada partícula é atualizada seguindo os valores da melhor solução encontrada pela partícula ao longo da busca, geralmente denominada pbest; e a melhor solução encontrada por todas as partículas, denominado de gbest (global best). No processo de busca, o comportamento de uma partícula depende de três possíveis escolhas: seguir seu próprio caminho ignorando as partículas vizinhas; seguir em direção à sua melhor posição já encontrada (pbest); Seguir em direção à melhor posição da melhor partícula vizinha (gbest). Após encontrar os valores do pbest e do gbest, cada partícula atualiza sua velocidade e posição. Como exemplos de algoritmos de Otimização por Nuvem de Partículas aplicados ao PCV, pode-se citar o algoritmo de Wang *et al.* (2003), Clerc (2004), Pang *et al.* (2004) e Machado e Lopes (2005).

A heurística ACO (*Ant Colony Optimization*) foi introduzida por Stützle e Dorigo (1999). Em seu trabalho, os autores resolvem 8 instâncias da biblioteca TSPLIB contendo de 198 à 1577 pontos, obtendo soluções com menos de 0,3% de excesso sobre as soluções ótimas, requisitando para isso poucos segundos para a menor instância e menos de 2 horas para a maior. Inauguram assim, uma nova e promissora metodologia para resolução do PCV.

Em 2000, Helsgaun (2000) faz uma revisão do algoritmo LK de Lin e Kernighan (1973), apresentando assim a versão LK-H. Esta nova versão foi capaz de resolver até à otimalidade a maioria das instâncias do TSPLIB consumindo menos de 1 segundo para problemas com menos de 100 cidades e cerca de 1 minuto para problemas com menos de 1.000 cidades. O algoritmo de Helsgaun só não se mostrou

eficiente para problemas com mais de 20.000 cidades por requerer excessivo tempo de processamento nestes casos.

Outras versões do LK podem ser encontradas na literatura, Gamboa *et al.* (2006) faz uma comparação entre seis dessas versões, propostas por diferentes grupos de pesquisadores. A análise envolveu a qualidade das soluções obtidas em cada versão e os tempos de processamento requeridos.

Possivelmente a versão mais contundente tenha sido proposta por Nguyen *et al.* (2007a), onde os autores se basearam no algoritmo de Helsgaun e realizaram duas modificações básicas: a primeira relacionada a estrutura de dados e a segunda para utilizar o movimento 5-Opt apenas no início do processo e à partir de então o movimento 3-Opt como básico no processo de busca LK.

Após as modificações, a versão de Nguyen *et al.* (2007a) conhecida como LK-NYYY chegou a ser aplicada em instâncias com até 1 milhão de cidades, com um tempo computacional razoável (GAMBOA *et al.*, 2006).

## 2.3 CONTRIBUIÇÕES EM ALGORITMOS GENÉTICOS

Em algum momento da educação básica fomos apresentados a Teoria de Darwin e não seria espantoso conhecermos termos tais como mutação, cruzamento, genótipo, fenótipo e evolução natural. Em sua teoria, Darwin mostrou que populações de indivíduos são capazes de evoluir ao longo das gerações, impulsionados por fatores aleatórios ou por influência de condições ambientais as quais estão sujeitos.

Para ele, as características de um indivíduo (normalmente comuns a outros indivíduos da população) são relevantes para determinar sua aptidão ao meio, além de serem transmitidas para seus descendentes. Quanto melhor for a herança que um indivíduo recebe de seus pais, maiores são as chances de sua prole perpetuar-se pelas gerações futuras.

Ao explorar os processos adaptativos de sistemas naturais e suas possíveis aplicabilidades em projetos de softwares e sistemas artificiais, o professor John Holland, da Universidade de Michigan nos EUA, formulou o algoritmo genético. Um AG imita regras presentes na natureza e observadas por Darwin. O trabalho publicado por Holland (1975) introduziu o método formalmente e permitiu abstrair e explicar rigorosamente os sistemas adaptativos naturais e ainda projetar softwares de sistemas artificiais que assegurassem mecanismos importantes de seleção natural (GOLDBERG, 1989).

Desde que foram desenvolvidos por Holland (1975), até meados de 2000,

diversos autores propuseram implementações de Algoritmos Genéticos para resolver o PCV, entretanto, os resultados desencorajavam os pesquisadores porque consumia-se muito tempo computacional e as soluções encontradas eram piores que as obtidas com outras heurísticas muito mais rápidas e já consagradas. No entanto, Nguyen *et al.* (2007b) apresentaram um AG híbrido com a heurística 5-Opt que foi capaz de obter resultados melhores que o LK-H de Helsgaun.

O AG híbrido de Nguyen *et al.* (2007b) foi capaz de encontrar a solução ótima mais vezes que o LK-H para diversas instâncias do TSPLIB, contudo o tempo computacional foi expressivamente maior, tanto que os experimentos foram realizados em um cluster com 10 computadores.

Liu e Zeng (2009) propuseram um AG com aprendizado por reforço, também apresentando tempos de processamento bem maiores que o LK-H, mais uma vez com resultados superiores no sentido de obter boas soluções ou mesmo a solução ótima com maior frequência.

### 2.3.1 RESULTADOS TEÓRICOS E O TEOREMA DO ESQUEMA

Apesar do bom desempenho numérico dos AGs para uma variedade de problemas, não há uma teoria geral que explique de forma completa como e por que esta técnica funciona. Desta forma, desde os primeiros trabalhos que empregam AGs, o objetivo é encontrar regras empíricas para melhorar seu desempenho (STEMMER, 2014). Na tentativa de produzir algum resultado teórico para AGs, algumas hipóteses bem aceitas foram propostas em Holland (1975) com o Teorema do Esquema e em Goldberg (1989) com a Hipótese dos Blocos Construtivos.

Segundo Holland (1975), com o passar das gerações, soluções boas tendem a compartilhar parte de seus cromossomos, chamadas de “esquemas”. Assim, bons esquemas tendem a se multiplicar exponencialmente ao longo das gerações, o que explica o desempenho de um AG.

Alguns autores assumem que um bom esquema deve ser curto, pois esquemas longos são mais suscetíveis aos operadores de cruzamento, sendo constantemente quebrados.

Por vezes pode ocorrer a chamada carona, ou seja, um esquema ruim aparecer em indivíduos contendo bons esquemas e prosperar ao longo das gerações por pegar carona com os bons esquemas.

Para Goldberg (1989), o poder dos AGs reside em ser capaz de encontrar “bons blocos construtivos”. Os blocos construtivos são padrões com comprimento

de definição pequeno, constituídos de posições que trabalham bem em conjunto e aumentam a aptidão de um Indivíduo.

## 2.4 PROCESSAMENTO EM PARALELO E MÉTODOS HÍBRIDOS

Em 2013, Honda, Nagata e Ono (2013) apresentam um AG com processamento em paralelo, encontrando as melhores soluções da literatura para problemas com até 180.000 cidades. Os autores concluem que o processamento em paralelo foi capaz de reduzir o tempo de execução em até 20 vezes mantendo a qualidade das soluções encontradas. Foram resolvidos PCVs com mais de 100.000 cidades, onde os autores conseguiram reduzir os menores trajetos já encontrados na literatura para as instâncias vangogh120K e courbet180K contendo 120.000 e 180.000 cidades, respectivamente.

Estas instâncias fazem parte de uma coleção presente em Cook (2014) e conhecida como PCVs Artísticos por ter sido construída sobre quadros de artistas mundiais famosos, tais como a obra Monalisa de Da Vinci. Nas observações finais do artigo, é mencionado que serão realizados trabalhos futuros na tentativa de obter os melhores trajetos da literatura para outras instâncias ainda não testadas.

Ghorpade e Kamalapur (2014) propõe um AG com processamento em paralelo, contudo com uma filosofia diferente da apresentada por Honda, Nagata e Ono (2013). Neste trabalho, os autores realizam o processamento em paralelo utilizando os núcleos de processamento da placa de vídeo do computador, ou seja, utilizam a GPU em vez de usar a CPU como realizado por Honda, Nagata e Ono (2013). Esta modificação pode acelerar ainda mais o AG, uma vez que a GPU possui centenas de processadores enquanto que a CPU somente cerca de uma dezena deles, com os recursos tecnológicos disponíveis atualmente.

Wang (2014) propõe um AG híbrido com duas estratégias de busca local mostrando que seu desempenho é superior ao AG clássico. Para comparação, o método foi aplicado em diversas instâncias da biblioteca TSPLIB.

Tsai *et al.* (2014) apresentam um AG de alto desempenho em termos de tempo de execução, tanto para o AG quanto para suas variações, o que implica em melhorar os tempos de AGs híbridos. Para isso a estratégia foi construir uma rotina que avalia e guarda trechos de cromossomo comuns a toda população ao longo das gerações, usando essas informações para evitar indivíduos não promissores. Nos testes foram usadas instâncias da biblioteca TSPLIB.

Atualmente percebe-se esforços por parte dos pesquisadores para propor hi-

bridizações eficientes para os AGs. Apesar de existirem outras metodologias com bom desempenho para a resolução do PCV, as propostas de hibridização presentes na literatura mais recente que tratam de AGs sugerem que seu uso é muito promissor, os argumentos a seguir apresentam uma justificativa parcial:

- A possibilidade de aproveitar o potencial de heurísticas consagradas na literatura em suas rotinas internas, seja para gerar a população inicial ou auxílio aos operadores genéticos.
- O fato dos AGs possuírem estrutura propícia a execução em paralelo, ação que está sendo explorada devido aos recentes avanços no campo da informática.
- Possuir a propriedade de explorar simultaneamente múltiplos pontos do espaço de soluções de um problema.
- As modificações em um AG para modelar variações do problema original são muito fáceis de implementar, diferentemente de outras heurísticas.
- Não requer premissas tais como: linearidade, convexidade e diferenciabilidade, entre outras.

### 3 MATERIAL E MÉTODO

Neste capítulo é apresentado o problema que é objeto de estudo desta pesquisa e algumas de suas aplicações reais mais comuns. Em seguida descreve-se as principais técnicas de solução disponíveis na literatura até o momento.

#### 3.1 O PROBLEMA DO CAIXEIRO VIAJANTE

O termo “Caixeiro Viajante” foi empregado pela primeira vez no meio científico em 1930, cujo precursor é desconhecido, porém o principal responsável por sua popularização foi Merrill Flood na tentativa de propor um desafio intelectual semelhante ao da Teoria dos Jogos (APPLEGATE *et al.*, 2007; SOUZA, 1997).

Diversos autores mencionam alguns aspectos motivadores para o estudo do PCV, entre eles estão a facilidade com que o problema é descrito, a dificuldade em resolvê-lo por ser NP-Completo e sua vasta aplicabilidade (KARP, 1975).

Para definir matematicamente o PCV devemos considerar um Grafo  $G(N, E)$  em que  $N = \{1, 2, \dots, n\}$  é o conjunto de nós ou vértices,  $E = \{1, 2, \dots, m\}$  é o conjunto de arestas de  $G$  e uma matriz  $C = [c_{ij}]$  tal que  $c_{ij}$  representa o custo associado a aresta que liga os vértices  $i$  e  $j$ . O problema consiste em determinar o menor ciclo hamiltoniano do Grafo  $G$ , sendo que o tamanho do ciclo é dado pelo somatório dos custos das arestas que o compõem (PRESTES, 2006).

Se usarmos os termos “cidades” ao invés de vértices e “estradas” ao invés de arestas o problema poderia ser reescrito como: determinar o trajeto mais curto de uma sequência de visitas partindo de uma das cidades, visitando todas as outras, uma única vez e, ao final, retornando ao ponto de partida.

Na Figura 1 é apresentado um exemplo com 48 cidades em que se assume haver um caminho entre quaisquer pares de cidades. O custo considerado é a distância euclidiana entre elas. Na Figura 1(a) consta apenas a localização cartesiana das cidades e na Figura 1(b) a solução ótima.



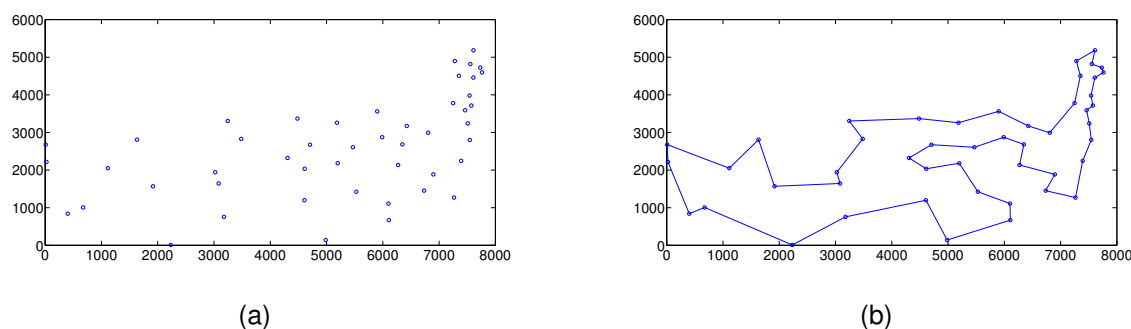


FIGURA 1: Exemplo de PCV com 48 cidades  
 FONTE: Instância att48 presente em Reinelt (1990)

### 3.2 APLICAÇÕES REAIS DO PCV

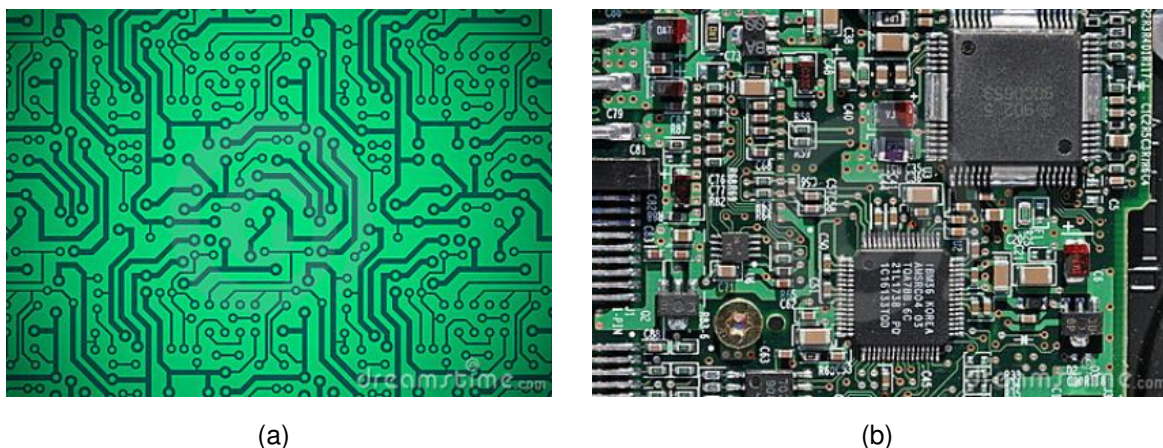
Uma motivação para o estudo de um problema está relacionada com sua aplicabilidade em situações reais. A seguir serão apresentadas algumas destas aplicações para o PCV. Cerca de uma dúzia de outras aplicações nas mais variadas áreas podem ser encontradas em Cook (2014).

#### 3.2.1 PERFURAÇÃO DE PLACAS DE CIRCUITO IMPRESSO

A otimização no processo de perfuração de placas de circuito impresso (PCI) é uma aplicação direta do PCV e desempenha um papel importante na linha de produção, por proporcionar consideráveis ganhos no tempo necessário para a realização desta tarefa.

As PCIs constituem-se de um sistema contendo componentes eletrônicos fixados na placa por furos e trilhas ligando-os. Na etapa de confecção destes furos, normalmente um equipamento se desloca sobre a placa realizando a tarefa, o que pode onerar o processo em termos de tempo, se o processo de deslocamento não estiver otimizado.

Um estudo mais detalhado para a linha de produção de uma grande indústria de componentes eletrônicos pode ser encontrado em Grötschel, Jünger e Reinelt (1991). Na Figura 2, duas imagens de placas distintas retratam alguns dos componentes comuns em uma placa. A Figura 2(a) mostra os furos e trilhas que fixam os componentes restantes na placa, mostrados na Figura 2(b).



(a)

(b)

FIGURA 2: Exemplo de Placas de Circuito Impresso

FONTE: <https://www.google.com.br/search?q=placas+de+circuito+impressos>

### 3.2.2 ANÁLISE DE ESTRUTURAS DE CRISTAIS

Neste processo, um difratômetro de raio-X é usado para obter informações sobre a estrutura do cristal. Para este propósito um detector mede a intensidade de reflexão dos raios-X em diferentes posições sobre o cristal. Como o resultado do experimento não depende da sequência em que são realizadas as medições, pode-se associá-las a um PCV determinando a sequência cujo tempo é menor, otimizando o processo (GRÖTSCHER; JÜNGER; REINELT, 1991). Para maiores informações acerca desta aplicação pode-se consultar Bland e Shallcross (1989).

### 3.2.3 MANUTENÇÃO DE SISTEMAS DE MOTORES DE TURBINA A GÁS

Esta aplicação foi relatada por Plante, Lowe e Chandrasekaran (1987) e ocorre quando motores de turbina a gás de aeronaves passam por manutenção. Para garantir um fluxo de gás uniforme pelas turbinas existem os chamados bicos de torção que são conjuntos situados em cada câmara da turbina. Esta montagem consiste basicamente de um número de bicos de palhetas guia fixadas em torno do seu perímetro.

Todas as palhetas têm características individuais e seu correto posicionamento pode proporcionar benefícios substanciais (redução das vibrações, aumentando a uniformidade do fluxo e reduzindo o consumo de combustível). O problema de colocação das palhetas na melhor posição possível pode ser modelado como um PCV com uma função objetivo especial.

### 3.2.4 MANIPULAÇÃO DE MATERIAIS EM UM ARMAZÉM

Segundo Ratliff e Rosenthal (1983) em geral existem diversos itens alocados no armazém das empresas e chegam pedidos de separação de subconjuntos desses itens que serão despachados para a linha de produção e formarão produtos finais comercializados pela empresa. Alguns veículos ou mesmo pessoas ficam encarregados de realizar a coleta desses itens.

A relação com o PCV é imediata, os locais de armazenamento dos itens correspondem aos nós do grafo enquanto que a distância entre dois nós é determinada pelo tempo necessário para se deslocar do endereço de um item para outro. O problema de encontrar um caminho mais curto para a sequência de coletas com um tempo mínimo de recolhimento pode ser resolvido como um PCV. Maiores detalhes desta aplicação pode ser encontrada em Dal (1992).

### 3.3 MÉTODOS EXATOS PARA RESOLUÇÃO DO PCV

A representação de um problema real por meio de uma linguagem matemática adequada constitui-se em um dos passos iniciais para sua resolução. Na sequência escolhe-se uma metodologia a ser aplicada. É bem difundido na literatura que as metodologias possíveis distinguem-se em pelo menos duas grandes categorias: os métodos exatos e os heurísticos. Especificamente para o caso do PCV existe uma gama de algoritmos, tanto heurísticos quanto exatos (PRESTES, 2006).

Quando se opta por um algoritmo exato, que utiliza a modelagem inteira, o pesquisador está sujeito a vantagens e desvantagens. Segundo Dumitrescu e Stutzle (2003), uma das vantagens é poder provar que soluções ótimas podem ser encontradas, quando o tempo de processamento permite que o algoritmo seja executado até atingir a condição de otimalidade. Outra vantagem é a possibilidade de obter limites inferiores e superiores para uma solução ótima desconhecida, uma vez que estes métodos ramificam o espaço de busca em níveis estruturados em função da qualidade das soluções (DUMITRESCU; STUTZLE, 2003).

Ainda durante a execução de um algoritmo exato, baseado na ramificação do espaço de busca, é possível abortá-lo precocemente e obter-se com esta ação, os limites inferiores e superiores, respectivamente, pelo melhor *bound* e a melhor solução encontrados. Por fim, os métodos exatos ainda permitem que regiões não promissoras do espaço de busca sejam descartadas.

Quanto às desvantagens, o tamanho das instâncias que podem ser resolvidas utilizando um método exato é limitado, seja pelo custo computacional expresso pelo

tempo de processamento elevado ou por requerer excessiva quantidade de memória.

Em Dumitrescu e Stutzle (2003) são mencionados alguns métodos exatos para solução do PCV, tais como: *Branch and Bound*, *Branch and Cut*, *Branch and Price*, Relaxação Lagrangeana e Programação Dinâmica.

Nesta seção são relatados os dois primeiros desses métodos. Indica-se Laporte (1992) como leitura complementar para mais detalhes acerca das abordagens exatas. Para iniciar a análise, será apresentada uma das formulações matemáticas possíveis para o PCV.

### 3.3.1 FORMULAÇÃO EXATA DO PCV

Em Bodin *et al.* (1983) são discutidas algumas formulações matemáticas para o PCV. Dentre estas, optou-se por apresentar neste trabalho a formulação dada a seguir:

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^N c_{ij} \times x_{ij} \quad (1)$$

s.a :

$$\sum_{i=1}^N x_{ij} = 1 \quad (\forall j = 1, \dots, N) \quad (2)$$

$$\sum_{j=1}^N x_{ij} = 1 \quad (\forall i = 1, \dots, N) \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad (\forall i, j = 1, \dots, N) \quad (4)$$

$$X = x_{ij} \in S \quad (\forall i, j = 1, \dots, N) \quad (5)$$

Onde  $c_{ij}$  e  $x_{ij}$  são, respectivamente, os custos e as variáveis de decisão associados com a tarefa de deslocar-se da cidade  $i$  para  $j$ .

A função objetivo (1) representa o custo total a ser minimizado. O conjunto de restrições (2) e (3) asseguram que cada cidade  $i$  será designada a exatamente uma cidade  $j$ . O conjunto (4) garantem que as variáveis  $x_{ij}$  assumirão os valores 0 ou 1.

O conjunto S, em (5) é qualquer grupo de restrições que impeçam a formação de subrotas, ou seja, conduzem a solução final a um ciclo hamiltoniano.

Em Bodin *et al.* (1983) são apresentadas algumas possibilidades para o conjunto S, tais como:

$$\bullet S = \left\{ (x_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \right\} \text{ para cada subconjunto próprio não vazio } Q \text{ de } N.$$

- $S = \left\{ (x_{ij}) : \sum_{i \in R} \sum_{j \in R} x_{ij} \leq |R| - 1 \right\}$  para todo subconjunto não vazio  $R \in \{2, 3, \dots, N\}$ .
- $S = \{ (x_{ij}) : y_i - y_j + Nx_{ij} \leq N - 1 \}$  para  $2 \leq i \neq j \leq N$  para os números reais  $y_i$ .

### 3.3.2 MÉTODO *BRANCH-AND-BOUND*

Trata-se de um método geral para resolução de problemas de otimização, proposto por Land e Doig (1960) cujos princípios norteadores são:

- Enumerar as soluções candidatas;
- Descarte de grande porção das soluções candidatas;
- Determinar limites superiores e inferiores usando heurísticas que produzam boas soluções ou por relaxações ao manipular uma ou mais restrições.

O esquema de enumeração divide o problema original em um número finito de subproblemas menores e mais fáceis de serem resolvidos. A solução destes subproblemas determina os limites superiores e inferiores e, a medida que o algoritmo é executado, todos os subproblemas com o valor da função objetivo acima do limite superior são descartados por não serem capazes de gerar soluções melhores que a melhor solução já conhecida.

Para a resolução de um PCV utilizando o método *Branch and Bound*, em regra geral, relaxa-se a exigência de variáveis binárias e enquanto houver variáveis da solução do problema relaxado em que ocorrer  $0 < x_{ij} < 1$  resolvem-se dois novos problemas com as restrições  $x_{ij} = 0$  em um deles e  $x_{ij} = 1$  em outro. Repete-se o procedimento anterior enquanto existir subproblemas por resolver ou variáveis na solução de algum subproblema que não assumiram valores binários (LAND; DOIG, 1960).

### 3.3.3 MÉTODO *BRANCH-AND-CUT*

O Método *Branch-and-Cut* de certa forma é uma variação do Método *Branch and Bound* e como apresentado em Prestes (2006), a ideia básica deste método consiste em encontrar uma relaxação linear de um problema inteiro ou binário, cujas soluções coincidam.

A cada etapa do processo de resolução do problema original, novas restrições são inseridas à sua versão relaxada linearmente para eliminar a ocorrência de variáveis fracionárias na solução final. O processo segue até que as variáveis da solução sejam binárias.

Este método tem sido aplicado com sucesso em várias classes de problemas de otimização combinatória, tais como: em PCVs (FISCHETTI; GONZALEZ; TOTH, 1997), em problemas de Steiner (LUCENA; BEASLEY, 1998) e em problemas de particionamentos de grafos (BRUNETTA; CONFORTI; RINALDI, 1997).

### 3.4 MÉTODOS HEURÍSTICOS PARA RESOLUÇÃO DO PCV

Métodos Heurísticos são algoritmos alternativos destinados a encontrar soluções aproximadas para um problema, mesmo que complexo, em tempos de processamento apropriados ao tomador de decisão. Contudo, perde-se a garantia de obtenção da solução ótima em oposição aos métodos exatos (SOUZA, 1997).

As primeiras heurísticas que foram propostas, geravam uma solução construída passo a passo pela inserção de cidades, considerando somente o maior benefício em cada nova inserção, sem avaliar um contexto global. Com o tempo, houve grande concentração de esforços por parte dos pesquisadores, que desenvolveram técnicas mais gerais, conhecidas como metaheurísticas, aplicáveis a um espectro maior de problemas (MICHALEWICZ; FOGEL, 2000).

Algumas das heurísticas construtivas frequentes na literatura são: Vizinho mais Próximo; Inserção do mais Próximo; Inserção do mais Barato; Inserção do mais Distante; Inserção Rápida; entre outras. Na Tabela 2 apresenta-se uma breve descrição de termos frequentes em heurísticas construtivas.

TABELA 2: Termos Frequentes em Heurísticas

Termo	Breve Descrição
Caminho	Sequência de vértices tal que cada um deles se une a uma única aresta para o vértice seguinte.
roteiro	Sinônimo para solução de um PCV.
rota	Sinônimo para roteiro.
subrota	trecho de uma rota.
ciclo hamiltoniano	Caminho que contém todos os vértices de um grafo.
grafo	Objeto matemático composto por vértices e arestas.
subgrafo	É um grafo formado por um subconjunto de vértices e arestas de outro grafo qualquer.
circuito	É um caminho que começa e acaba com o mesmo vértice.

FONTE: Autor(2015)

### 3.4.1 VIZINHO MAIS PRÓXIMO

Esta heurística constrói a solução passo a passo, por meio de uma solução parcial que inicia-se com um vértice aleatório e, a cada passo, insere-se um novo vértice por ser o vizinho mais próximo do último vértice inserido. O pseudo algoritmo está apresentado a seguir.

---

#### ALGORITMO: Vizinho mais Próximo

---

- Passo 1: Comece o caminho em um vértice qualquer.
- Passo 2: Encontre o vértice não pertencente ao caminho que esteja mais próximo do último vértice adicionado; adicione-o no final do roteiro em formação.
- Passo 3: Repita o passo 2 até que o caminho compreenda todos os vértices, e quando isto ocorrer una o último vértice ao primeiro fechando a rota.
- 

### 3.4.2 HEURÍSTICA DE MELHORIA K-OPT

O procedimento k-Opt é uma generalização da heurística de busca local 2-Opt proposta por Croes (1958). Um passo da busca 2-Opt apaga duas arestas de uma solução corrente, transformando-a em dois sub percursos. Na sequência, religa-se suas extremidades modificando a solução, caso haja ganho na função objetivo toma-se a nova solução como solução corrente e executa o procedimento novamente para outras duas arestas.

Durante a execução da heurística, se tomarmos outro valor de k maior que 2, temos então um procedimento k-Opt. Espera-se que quanto maior o valor de k escolhido, mais próximo se chega da solução ótima. Em um problema com n cidades se executarmos um procedimento n-Opt, teremos ao final a solução ótima porque avaliam-se todas as soluções possíveis.

Valores de k maiores que 5 não são praticados na literatura, uma vez que o tempo computacional para execução de uma heurística k-Opt sobre um problema com n cidades é da ordem de  $n^k$ , o que representa tempos exponenciais, à medida que n e k crescem (SOUZA, 1997).

Os valores de k mais frequentes na literatura são 2, 3 e algumas vezes 5. Sempre promovem consideráveis melhorias. Este procedimento pressupõe uma solução inicial normalmente obtida por outras metodologias, tais como heurísticas construtivas, Busca Tabu, *Simulated Annealing*, Algoritmos Genéticos, Redes Neurais, entre outras.

A heurística 3-Opt é clássica na literatura e promove melhorias em uma solução qualquer para o PCV (CROES, 1958), (JÜNGER; RINALDI; REINELT, 1995) e

(MATAI; SINGH; MITTAL, 2010). Para melhor detalhar o funcionamento da heurística k-Opt, um exemplo envolvendo 3-Opt é apresentado na Figura 3 ilustrando um passo da heurística.

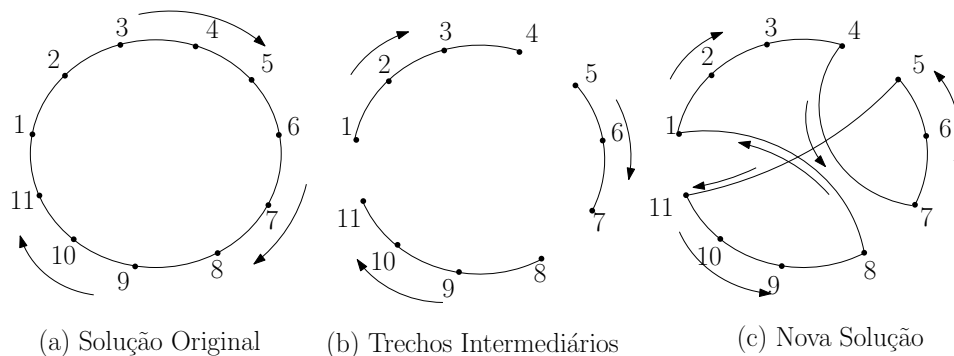


FIGURA 3: Passo de 3-Opt sobre um PCV

FONTE: Autor(2015)

A Figura 3(a) demonstra a solução imediatamente anterior a aplicação de um passo da heurística, a Figura 3(b) indicando as arestas apagadas pelo passo de 3-Opt, e por fim a Figura 3(c) que apresenta a solução após a reconexão dos trechos envolvidos.

### 3.4.3 ALGORITMO DE LIN E KERNIGHAN E SUAS VARIAÇÕES

As variações do algoritmo proposto por Lin e Kernighan (1973) conhecido como LK, representa a classe de heurísticas mais promissoras para o PCV simétrico. Ao longo dos anos diversas variações foram implementadas visando melhorar o desempenho deste algoritmo.

Atualmente, estas variações são as metodologias com melhor desempenho para instâncias de grande dimensão do PCV. Desde setembro de 2013, a melhor solução para o PCV Mundial, um problema com 1.904.711 cidades, foi obtida por uma versão revisada do algoritmo LK-H. Diversas outras variações do LK podem ser encontradas na literatura, tais como o algoritmo LK-ABCC de Applegate *et al.* (1999), o algoritmo LK-NYYY de Nguyen *et al.* (2007a), o algoritmo LK-N de Neto (1999) entre outros.

Estes algoritmos também realizam movimentos de melhoria k-Opt, cujo valor de k varia entre 2 e 5 e representam a quantidade de arestas de uma solução corrente a serem trocadas a cada iteração. As trocas seguem os passos apresentados no algoritmo LK que pode ser consultado em Prestes (2006).



### 3.5 ALGORITMO GENÉTICO

Nesta seção é discutido o AG, que é a metodologia fundamental para a construção do método proposto neste trabalho. Nas subseções seguintes são descritas as particularidades do método, que incluem os parâmetros necessários a uma boa implementação computacional.

#### 3.5.1 CARACTERÍSTICAS GERAIS DO MÉTODO

Segundo Borges (2003), para o emprego de um AG é necessário observar os seguintes componentes essenciais:

- A representação genética para soluções candidatas ou potenciais (codificação);
- Uma maneira de criar a população inicial de soluções candidatas ou potenciais;
- Uma função de avaliação que faz o papel da pressão do ambiente, classificando as soluções em termos de sua aptidão;
- O emprego de operadores genéticos e valores para os parâmetros usados pelo algoritmo, tais como tamanho da população, taxas de probabilidade da aplicação de operadores genéticos, entre outros.

A Tabela 3, adaptada de Borges (2003) apresenta uma analogia entre os conceitos usados em teoria da evolução e os usados em um AG.

TABELA 3: Termos utilizados em AGs e na Biologia

Termo	Biologia	AGs
Gen ou Gene	Unidade de hereditariedade transmitida pelo cromossomo que controla as características do organismo.	Parâmetro codificado no cromossomo, ou seja, um elemento do vetor que representa o cromossomo.
Genoma/Cromossomo	Conjunto completo de genes de um organismo. Genoma é o conjunto de vários cromossomos.	Estrutura de dados que codifica uma solução para um problema, ou seja, um cromossomo representa um único ponto no espaço de busca.
Fitness	Valor que indica o grau de aptidão do indivíduo ao meio que está inserido.	Fornece para cada indivíduo a medida de quão próximo a uma solução considerada satisfatória se encontra.
Indivíduo	Um membro da população.	Um indivíduo é formado pelo cromossomo (conjunto de genes) e por um valor de fitness.
Genótipo	Composição genética contida no genoma.	Informação contida no cromossomo.
Fenótipo	Características visíveis, como cor dos olhos, dos cabelos e outras características físicas.	Objeto, estrutura ou organismo constituído a partir das informações do genótipo representando a codificação do cromossomo.
Alelo	Formas alternativas do gene.	Valores que o gene pode assumir.

FONTE: Adaptado de Borges (2003)

Um AG atua de forma paralela sobre o espaço de busca, manipulando simultaneamente diversos pontos deste espaço por recombinação dos indivíduos da população corrente. O paralelismo lhe confere a possibilidade de escapar de ótimos locais enquanto outras técnicas, que atuam num único ponto, podem não ter o mesmo sucesso.

Os termos em inglês *Exploration*  $\times$  *Exploitation* ajudam no entendimento das potencialidades dos AGs. *Exploration* é a exploração que investiga áreas novas e desconhecidas no espaço de busca. Já *Exploitation* é a exploração que aproveita ao máximo o conhecimento obtido em pontos visitados do espaço de busca. Algoritmos de busca aleatória são especialistas em *Exploration* mas não fazem *Exploitation*. Por outro lado, métodos que usam o gradiente são bons em *Exploitation* mas fazem pouco *Exploration*. Os AGs realizam as duas ações simultaneamente, o que pode torná-lo mais efetivo (STEMMER, 2014).

O fluxograma apresentado na Figura 4 proporciona um melhor entendimento da estrutura de funcionamento de um AG.

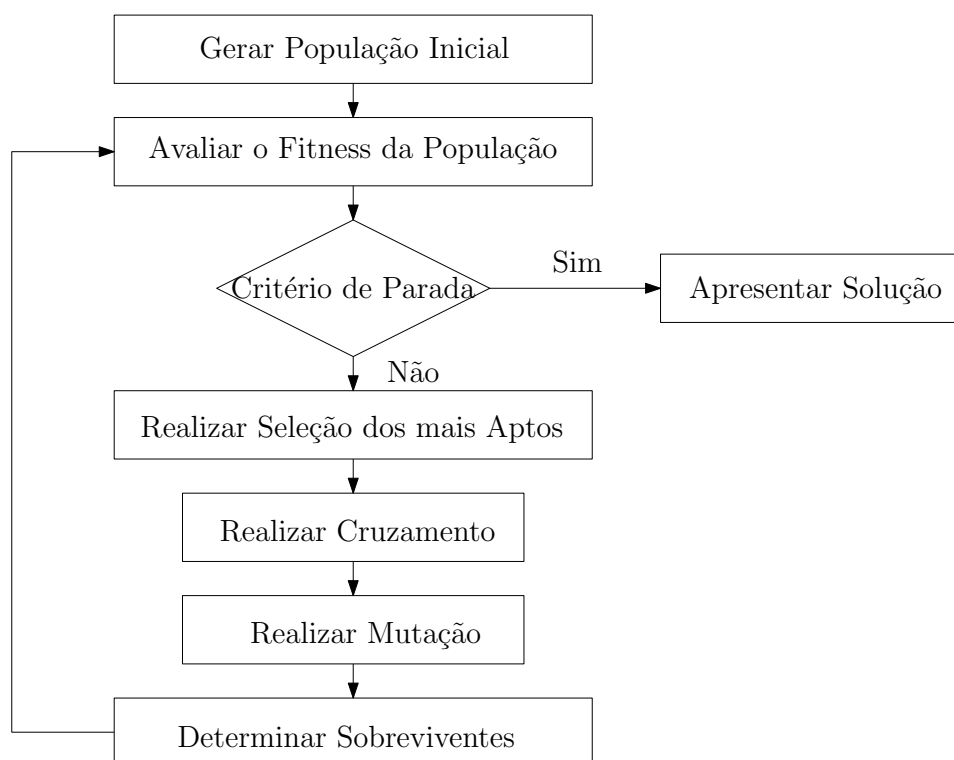


FIGURA 4: Passos de um AG Clássico

FONTE: Autor(2015)

### 3.5.2 REPRESENTAÇÃO DAS SOLUÇÕES NO ESPAÇO DE BUSCA

A busca por uma representação adequada para uma solução (ou um indivíduo como é conhecida nas terminologias de um AG) é uma etapa importante da implementação do AG. Normalmente, esta representação é dada por um vetor composto por números binários.

No entanto, cada problema pode requerer outras representações mais adequadas, tais como: vetor de inteiros, vetor de reais, matriz, entre outros. No caso do PCV, objeto do presente estudo, descrevem-se as possíveis representações por Caminho, Ordinal, Adjacência e por uma Matriz.

#### 3.5.2.1 REPRESENTAÇÃO POR CAMINHO

A representação por caminho é a mais natural possível. Por exemplo, uma solução para um PCV com 8 cidades, nesta representação, é obtida arranjando-as em um vetor com 8 posições. A posição que cada cidade figura no vetor representa a ordem em que ela será visitada.

A Figura 5 ilustra o caso em que a primeira cidade a ser visitada é a cidade 8, na sequência visita-se a cidade 2 e assim por diante até ocorrer a última visita na cidade 3 sendo que o retorno à cidade de origem fica implícito.

8	2	5	7	4	1	6	3
---	---	---	---	---	---	---	---

FIGURA 5: Representação por Caminho para um PCV

FONTE: Autor(2015)

Esta representação possui o inconveniente de duas configurações diferentes, representarem a mesma solução como mostrado na Figura 6(a) e (b). Entretanto, este problema pode ser contornado fixando-se sempre a mesma cidade como a inicial de qualquer roteiro (POTVIN, 1996).

8	2	5	7	4	1	6	3
---	---	---	---	---	---	---	---

(a) Primeira Configuração

1	6	3	8	2	5	7	4
---	---	---	---	---	---	---	---

(b) Segunda Configuração

FIGURA 6: Configurações Diferentes Para uma Mesma Solução

FONTE: Autor(2015)

#### 3.5.2.2 REPRESENTAÇÃO ORDINAL

Na representação ordinal, deve-se considerar um vetor canônico auxiliar que fornece as cidades para se construir uma solução passo a passo. Assim, a solução é

uma lista de posições que indica a ordem em que as cidades são retiradas do vetor canônico. Para a mesma solução da Figura 5 expõe-se a construção de sua representação ordinal passo a passo na Tabela 4.

TABELA 4: Construção da Representação Ordinal

Passos	Vetor Canônico								Representação Ordinal							
Passo 1:	1	2	3	4	5	6	7	8	8							
Passo 2:	1	2	3	4	5	6	7		8	2						
Passo 3:	1	3	4	5	6	7			8	2	4					
Passo 4:	1	3	4	6	7				8	2	4	5				
Passo 5:	1	3	4	6					8	2	4	5	3			
Passo 6:	1	3	6						8	2	4	5	3	1		
Passo 7:	3	6							8	2	4	5	3	1	2	
Passo 8:	3								8	2	4	5	3	1	2	1

FONTE: Adaptado de Potvin (1996)

Esta representação é útil para um AG que utiliza operador de cruzamento 1PX, pois gera soluções sempre factíveis após a aplicação do operador sobre duas soluções quaisquer (POTVIN, 1996).

### 3.5.2.3 REPRESENTAÇÃO POR MATRIZ ESPARSA

Nesta representação a solução é codificada numa matriz cujo elemento da linha  $i$  e coluna  $j$  é nulo se as cidades  $i$  e  $j$  não forem adjacentes e 1 se a cidade  $i$  precede a cidade  $j$  ou a cidade  $j$  precede a cidade  $i$  na solução. Para exemplificar apresenta-se o trajeto do exemplo da Figura 5 por meio de uma matriz esparsa na Tabela 5.

TABELA 5: Representação de uma Solução por Matriz Esparsa

0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	1
1	0	0	0	0	0	1	0
0	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0
0	1	1	0	0	0	0	0

FONTE: Autor(2015)

Codificar uma solução por matriz não é útil quando se implementa um AG, pois após aplicar os operadores de cruzamento ou mutação é frequente a necessidade de reparos para manter a solução factível, além do mais estes reparos consomem relativo esforço computacional. No entanto, esta representação será valiosa na discussão e implementação de algumas etapas do método híbrido tratado neste trabalho.

### 3.5.3 FUNÇÃO DE APTIDÃO

De acordo com Goldberg (1989), a função de aptidão é construída considerando-se as particularidades de cada problema e deve ser capaz de refletir a qualidade de um indivíduo em solucionar o problema. Normalmente, ela relaciona-se diretamente com a função objetivo a ser otimizada.

Para o PCV costuma-se considerar como função de aptidão o somatório de todas as distâncias percorridas para realizar o roteiro indicado por uma solução. Ichihara (2013) afirma que um dos problemas mais frequentes em um AG é a convergência prematura, que pode estar relacionado à dominância dos indivíduos com alto valor de aptidão nas populações iniciais.

Deste fato, surge a necessidade de modificar a função de aptidão para permitir a contribuição dos indivíduos menos aptos, através de vários mecanismos, dentre os quais temos o procedimento de escala ou o procedimento de *rank*. A ideia básica do procedimento de escala é limitar a competição entre os indivíduos nas iterações iniciais e estimulá-la progressivamente, através de uma mudança (geralmente linear) de escala na aptidão. Quanto ao procedimento de *rank*, a estratégia é numerar os  $N$  indivíduos da população crescentemente de 0 a  $N$  de acordo com seu *fitness* e fazer desta numeração suas novas aptidões.

### 3.5.4 POPULAÇÃO INICIAL

Não há relatos na literatura, de uma regra geral para a construção da população inicial. Pode ocorrer de uma população inicial contendo indivíduos mais aptos convergir para uma solução final não tão boa quanto a de outra população inicial contendo indivíduos menos aptos. Normalmente os autores optam entre dois caminhos para gerar a população inicial: randomicamente ou então utilizam heurísticas consagradas que consideram particularidades do problema para prover boas soluções iniciais.

Para o PCV, por exemplo, Ismkhan e Zamanifar (2013) comparam o impacto sobre a velocidade e a precisão do AG e, para tal, utilizou-se uma população inicial

gerada randomicamente. Panwar e Gupta (2013) não apresentam resultados contundentes em sua implementação mas também utilizam uma população inicial randômica. Em Deep e Mebrahtu (2011), a geração da população inicial foi realizada com o auxílio da heurística do vizinho mais próximo.

#### 3.5.4.1 MÉTODO DO VIZINHO MAIS PRÓXIMO ADAPTADO

O método proposto nesta tese, gera a população inicial por meio de um algoritmo simples que mescla o algoritmo do Vizinho Mais Próximo com o conceito de gerar uma solução totalmente aleatória, cujo pseudocódigo é apresentado a seguir:

---

##### ALGORITMO: Gerar População Inicial

---

Dados de Entrada: **TamPop**, **QtdeCidades**, **ParametroB**

Criar uma **Lista** de vizinhança contendo **ParâmetroB** vizinhos para cada cidade

Enquanto o tamanho da população não atingir **TamPop**. Faça

Selecione aleatoriamente uma cidade para iniciar o roteiro

Enquanto o roteiro for menor que **QtdeCidades**. Faça

Selecione aleatoriamente a nova cidade do roteiro, entre opções da **Lista** de vizinhança  
atualizar a **Lista**

Fim

Insira o roteiro em **População**

Fim

Dados de Saída: **População**

---

Este pseudocódigo inicia a construção de um indivíduo partindo de uma cidade escolhida aleatoriamente, assim como no método do vizinho mais próximo. Nos passos seguintes, em vez de inserir no roteiro a cidade que estiver mais próxima da última cidade previamente inserida, faz-se um sorteio entre um conjunto de vizinhos mais próximos desta última cidade recém incluída para determinar qual a próxima cidade a ser inserida. A quantidade de vizinhos que participa do sorteio a cada passo é determinada pelo “ParâmetroB”.

Nas figuras de 7 a 11 é mostrado um exemplo de construção de uma solução para o PCV com 10 cidades sendo fixado “ParametroB igual” a 2. Na Figura 7 a cidade 1 foi sorteada aleatoriamente para iniciar o trajeto. Na sequência, define-se uma região contendo seus dois vizinhos mais próximos e escolhe-se entre estes, de forma aleatória, a cidade 6 para ser inserida no roteiro parcial. O mesmo processo é realizado para a cidade 6 na etapa 2 conduzindo à inserção da cidade 8 no trajeto.

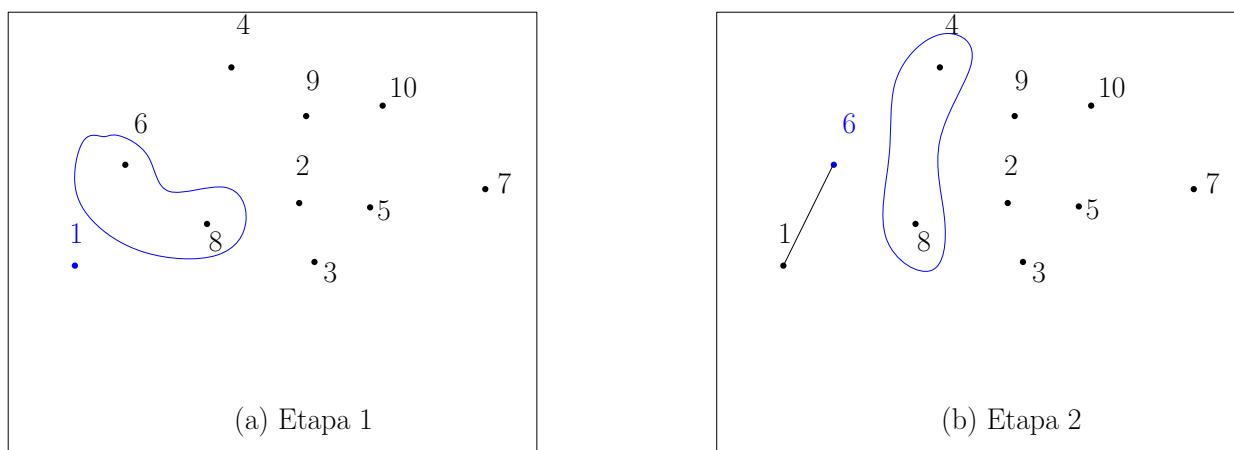


FIGURA 7: Etapas 1 e 2 da Geração da Solução

FONTE: Autor(2015)

Nas imagens da Figura 8 insere-se a cidade 2 a partir da vizinhança da cidade 8 e na sequência a cidade 3 a partir da vizinhança de 2.

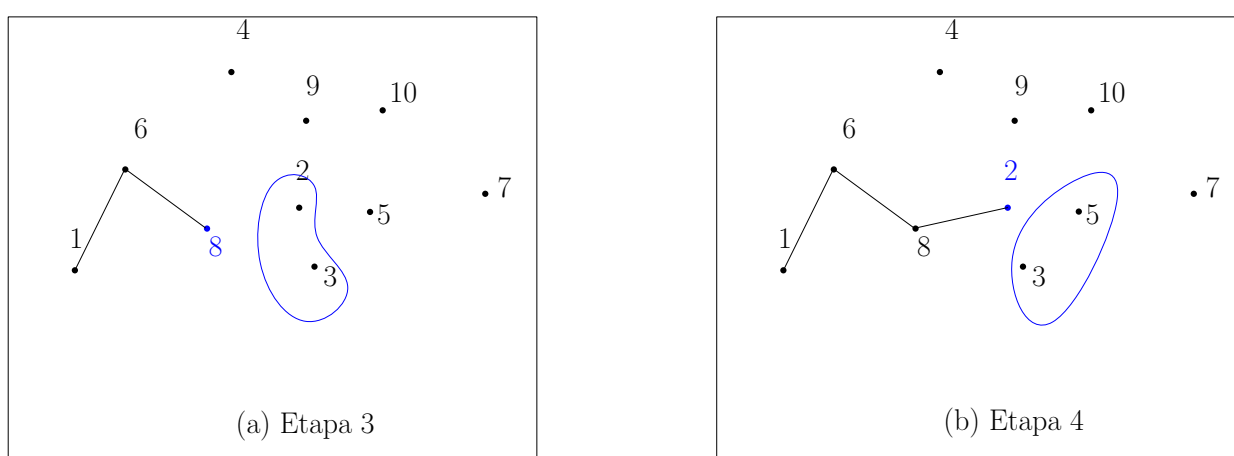


FIGURA 8: Etapas 3 e 4 da Geração da Solução

FONTE: Autor(2015)

Na etapa 5 da Figura 9 insere-se a cidade 5 a partir da vizinhança da cidade 3. Para a etapa 6, os dois vizinhos mais próximos da cidade 5 são as cidades 10 e 7. Entre estas opções, insere-se aleatoriamente a cidade 10.

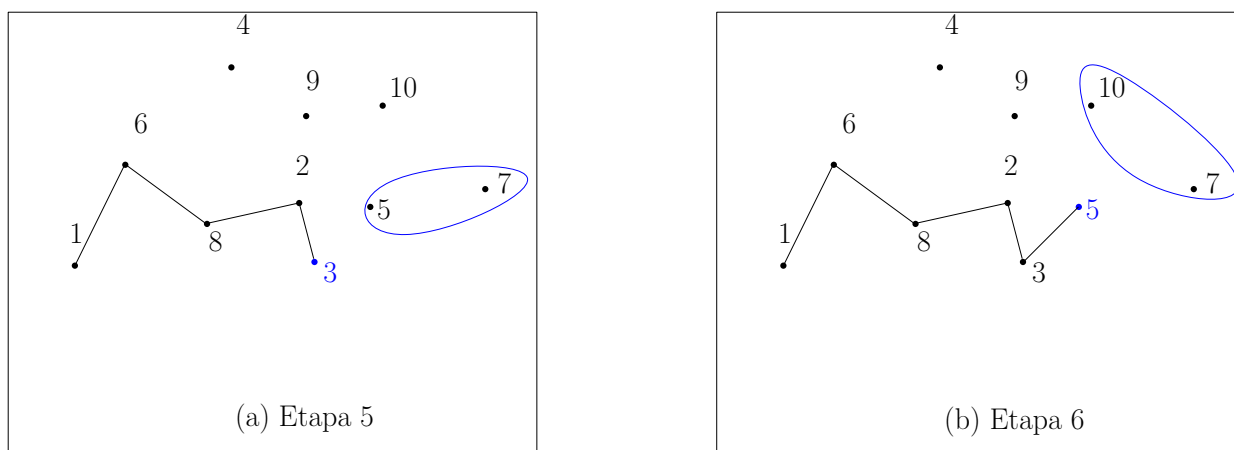


FIGURA 9: Etapas 5 e 6 da Geração da Solução

FONTE: Autor(2015)

As etapas 7 e 8 são mostradas na Figura 10 e ilustram a inserção das cidades 7 e 4.

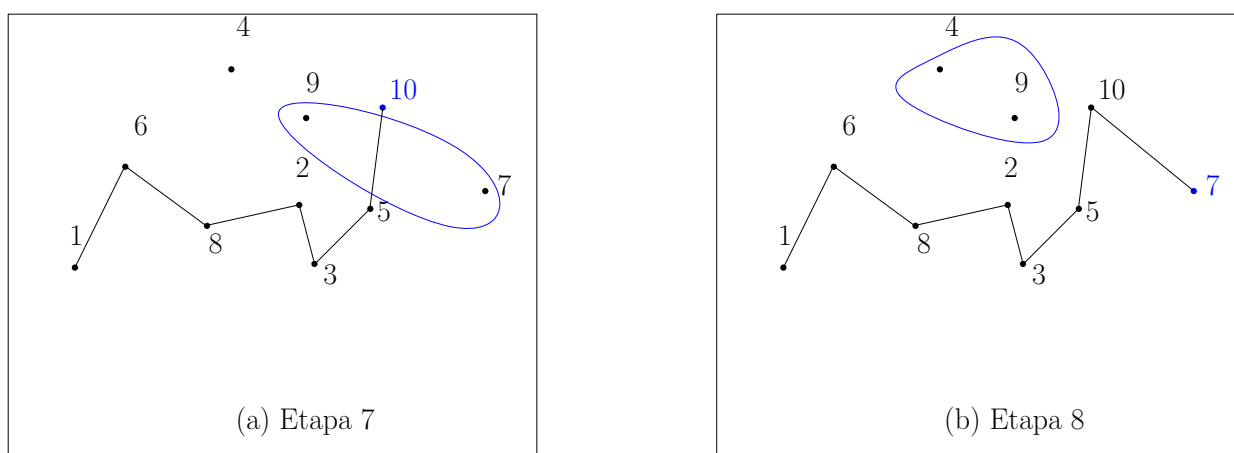


FIGURA 10: Etapas 7 e 8 da Geração da Solução

FONTE: Autor(2015)

A etapa 9, onde insere-se a cidade 9, bem como o retorno à cidade de origem que ocorre na etapa 10, finalizando-se a construção da solução, são apresentadas na Figura 11.



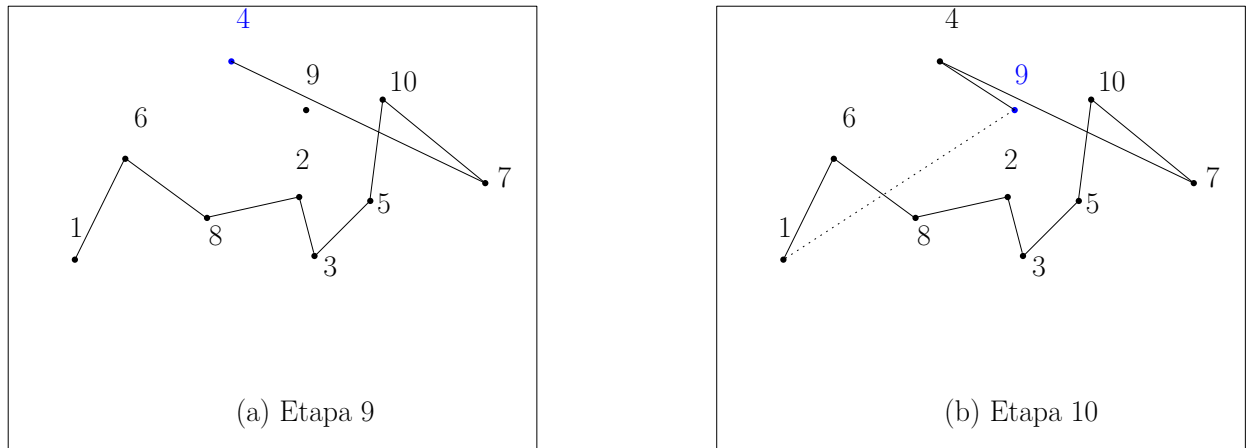


FIGURA 11: Etapas 9 e 10 da Geração da Solução

FONTE: Autor(2015)

### 3.5.5 MÉTODOS DE SELEÇÃO

O operador de seleção de um AG é uma analogia com a seleção biológica natural de Darwin. A cada geração determina-se quais indivíduos estarão sujeitos aos operadores genéticos de cruzamento e mutação. Uma boa escolha do método é fundamental para conduzir a busca no AG. Ao selecionar os indivíduos que irão gerar os novos descendentes, implicitamente, o método contribui para incentivar ou restringir a busca em determinadas regiões do espaço de soluções, sejam elas mais promissoras ou não.

Os indivíduos com melhor *fitness* ou aptidão tendem a ser selecionados para produzirem descendentes para as gerações futuras, assim como ocorre na natureza. Infelizmente este processo pode conduzir a um dilema, Se os indivíduos com melhor *fitness* forem frequentemente selecionados e estes forem muito mais aptos que os demais, pode ocorrer convergência prematura, enquanto que se forem privilegiados os indivíduos com baixo *fitness* a convergência pode ser lenta (CUNHA; TAKAHASHI; HENGGELE, 2013).

Existem diversos procedimentos de seleção presentes na literatura, tais como os métodos: Roleta, Boltzmann, Escalada Sigma, Torneio, entre outros.

Para realizar a seleção pelo método da roleta, inicialmente calcula-se a probabilidade de reprodução pela equação:

$$P_i = \frac{f_i}{\sum_{i=1}^N f_i} \forall i \in \{1, 2, \dots, N\} \quad (6)$$

onde  $P_i$  é a probabilidade do indivíduo  $i$  ser sorteado para reproduzir-se,  $N$  é o tamanho da população e  $f_i$  é a aptidão do indivíduo  $i$ .

O método da roleta consiste em distribuir setores de uma roleta entre os indivíduos da população, proporcionalmente aos valores de seus *fitness*, para na sequência girar a roleta e selecionar um indivíduo. Este método promove a convergência prematura em populações que apresentam alguns indivíduos muito melhores que o restante.

Para o método do torneio, inicialmente seleciona-se aleatoriamente um subconjunto da população, contendo dois indivíduos que irão competir entre si e aquele que possuir maior aptidão é escolhido como um dos pais ao qual serão aplicados os operadores genéticos, na sequência repete-se o procedimento a fim de obter o segundo pai. De acordo com Reyes (2011), o torneio e a roleta são os métodos de seleção mais utilizados em AGs para o PCV.

Neste trabalho propõe-se um novo método, cujo objetivo é incentivar a diversidade genética na população e ao mesmo tempo evitar que os indivíduos sejam selecionados para o cruzamento apenas por seus *fitness*, em contraposição ao que ocorre com a maioria dos métodos de seleção. Cada casal de indivíduos que sofrerão a ação dos operadores são determinados da seguinte forma:

- Inicialmente aplica-se o método do torneio para determinar um casal potencial;
- Na sequência executa-se a rotina Aplicar Critério de similaridade (ACS) para verificar se o cruzamento ocorrerá de fato.

A similaridade é obtida pela contagem simples da quantidade de arestas que ambas as soluções possuem em comum, para facilitar este processo inicialmente representam-se as soluções na forma de matriz e, em seguida, basta realizar um produto matricial ponto a ponto.

O produto ponto a ponto é equivalente ao produto das posições correspondentes entre duas matrizes, ou seja, dadas as duas matrizes:

$$A = [a_{ij}] \text{ e } B = [b_{ij}] \quad (7)$$

então o produto ponto a ponto entre A e B, definido por  $C = [c_{ij}]$  é tal que:

$$c_{ij} = a_{ij} * b_{ij} \quad \forall i, j \quad (8)$$

A Figura 12 ilustra a operação de cálculo da similaridade para duas soluções quaisquer de um PCV dado. No caso em questão o valor da similaridade obtido foi 6, ou seja, há 6 arestas comuns às duas soluções apresentadas.

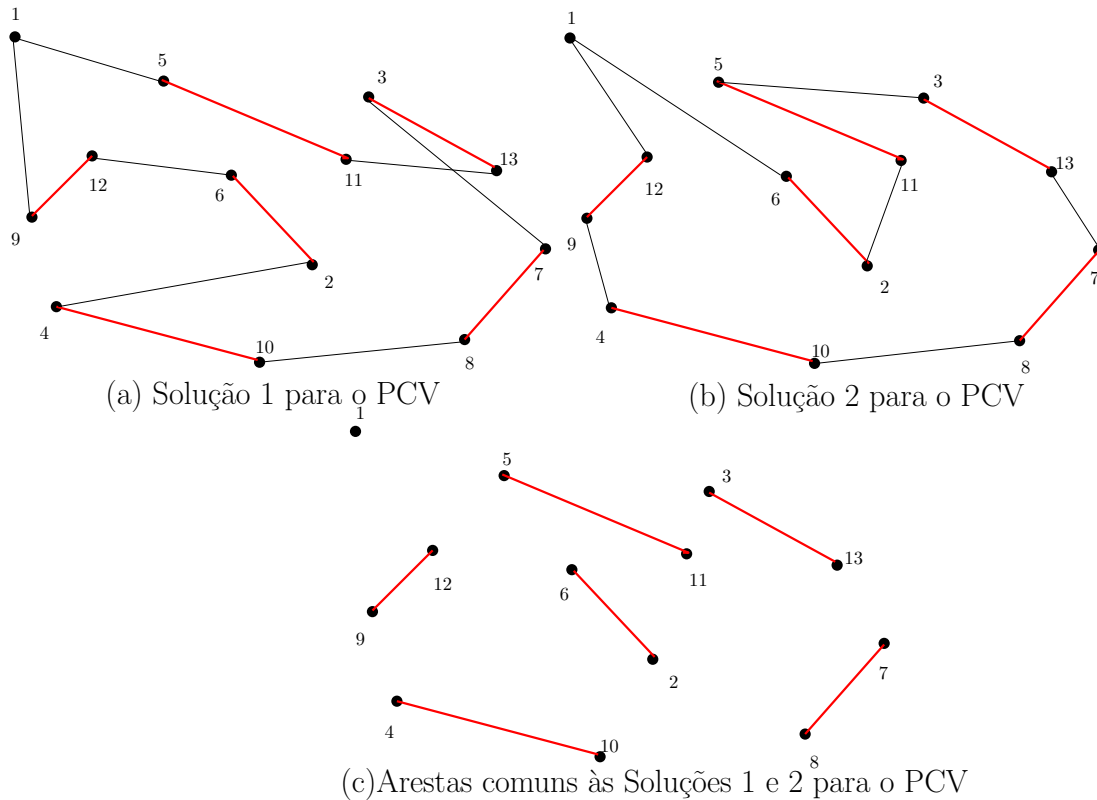


FIGURA 12: Cálculo da Similaridade Entre duas Soluções

FONTE: Autor(2015)

A similaridade assume valores inteiros entre 0 e a quantidade de cidades envolvidas no PCV. Para o exemplo da Figura 12 o maior valor obtido seria 13, caso as soluções 1 e 2 fossem idênticas.

A rotina ACS auxilia o método do torneio no processo de seleção dos indivíduos. Após dois indivíduos serem pré-selecionados pelo método do Torneio e executado a rotina ACS obtendo a “similaridade” entre elas, a dupla é submetida às seguintes análises:

- Se  $Iteracao \leq LimiteA \times CriterioParada$ , então descartar candidatos com similaridade acima de  $LimiteA\%$  do número de vértices do problema;
- Se  $LimiteA \times CriterioParada < Iteracao \leq LimiteB \times CriterioParada$ , então descartar candidatos com similaridade acima de  $LimiteB\%$  do número de vértices do problema;
- Se  $LimiteB \times CriterioParada < Iteracao \leq LimiteC \times CriterioParada$ , então descartar candidatos com similaridade acima de  $LimiteC\%$  do número de vértices do problema
- Se  $Iteracao > LimiteC \times CriterioParada$ , então não avaliar a similaridade.

### 3.5.6 OPERADORES GENÉTICOS - CRUZAMENTO

Os operadores genéticos são os responsáveis por transmitir a informação genética dos indivíduos a seus descendentes. Em geral, os operadores de cruzamento partem de dois indivíduos da população e por recombinação geram um ou dois novos indivíduos com características herdadas de seus pais. Em alguns tipos de problemas, após o cruzamento pode ser necessário um passo extra de correções para tornar os filhos factíveis.

A estrutura de dados escolhida para representar um indivíduo determina quais operadores poderão ser utilizados. Potvin (1996) apresenta uma revisão de literatura sobre diversos operadores para o PCV representado por caminho, tais como PMX, OX, CX, OBX e PBX. O estudo de Larranaga *et al.* (1999), apresentado na Tabela 6, lista os principais operadores genéticos utilizados em diversas estruturas do PCV.

TABELA 6: Principais Operadores Genéticos até 1999

Representação	Operador	Autor
Binário		
	Clássico + reparação (1PX)	Lidd (1991)
Caminho		
	<i>Simple Inversion Mutation</i> (SIM)	Holland (1975)
	<i>Sorted Match Crossover</i>	Brady (1985)
	<i>Partially - Mapped Crossover</i> (PMX)	Goldberg and Lingle (1985)
	<i>Order - Crossover</i> (OX)	Davis (1985)
	<i>Cycle Crossover</i> (CX)	Oliver <i>et al</i> (1987)
	<i>Heuristic Crossover</i> (HX)	Grefenstette (1987b)
	<i>Maximal Preservative Crossover</i> (MPX)	Mühlenbein <i>et al</i> (1988)
	<i>Insertion Mutation</i> (ISM)	Fogel (1988)
	<i>Edge Recombination Crossover</i> (ERX)	Whitley <i>et al</i> (1989)
	<i>Voting - Recombination Crossover</i> (VR)	Mühlenbein (1989)
	<i>Exchange Mutation</i> (EM)	Banzaf (1990)
	<i>Inversion Mutation</i> (IVM)	Fogel (1990)
	<i>Seramble Mutation</i> (SM)	Syswerda (1991)
	<i>Order Based Crossover</i> (OBX)	Syswerda (1991)
	<i>Position Based Crossover</i> (POS)	Syswerda (1991)
	<i>Displacement Mutation</i> (DM)	Michalewicz (1992)
	<i>Alternating - Position Crossover</i> (AP)	Larrañaga <i>et al</i> (1996a)
Adjacência		
	<i>Alternating Edge Crossover</i>	Grefenstette (1985)
	<i>Subtour Chunks Crossover</i>	Grefenstette (1985)
	<i>Heuristic Crossover 1</i>	Grefenstette (1987b)
	<i>Heuristic Crossover 2</i>	Jog <i>et al</i> (1989)
	<i>Heuristic Crossover 3</i>	Suh and Van Gucht (1987)
		continua na próxima página

continuação da página anterior		
Representação	Operador	Autor
Ordinal	<i>Classical operators</i>	Grefenstette (1985)
	<i>Intersection Crossover Operator</i>	Fox and Mc Mahon (1987)
Matricial	<i>Union Crossover Operator</i>	Fox and Mc Mahon (1987)

FONTE: Adaptado de Larranaga *et al.* (1999)

### 3.5.6.1 OPERADOR PMX

Este operador é devido a Goldberg (1985) e constrói dois novos descendentes que preservam uma subsequência aleatória de um dos pais e mantêm a ordem e a posição de quantas cidades for possível do outro pai. Na Figura 13 indica-se os pontos de corte e as etapas para gerar um dos filhos com o operador PMX.

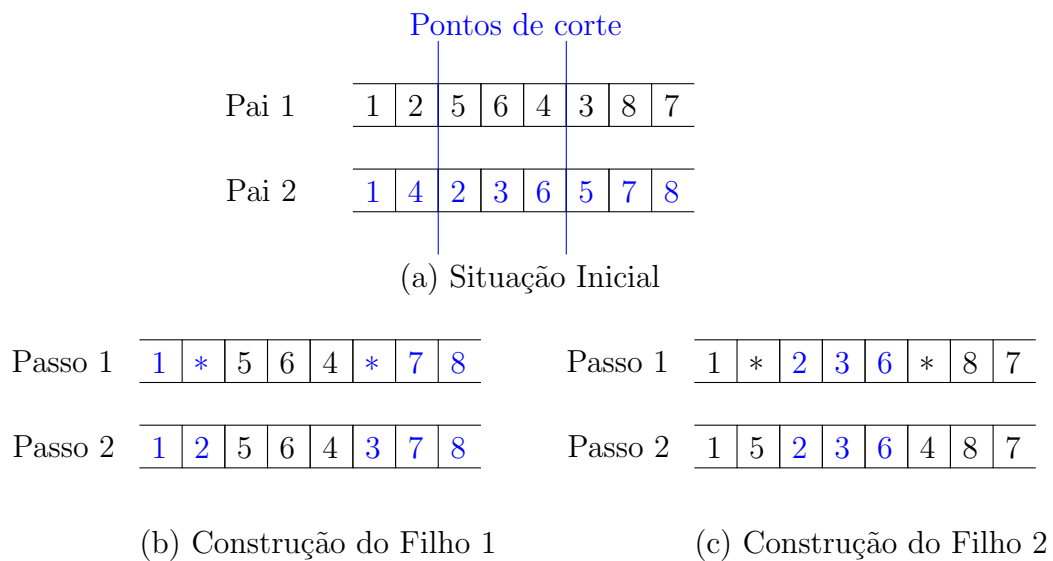


FIGURA 13: Construção de Descendentes com Operador PMX

FONTE: Autor(2015)

No Passo 1 da parte b) da Figura 13, selecionou-se o trecho interior aos pontos de corte do primeiro pai e na sequência as cidades exteriores aos pontos de corte do segundo pai preservando a ordem e a posição que elas aparecem. O símbolo \* substitui as cidades que não puderam ser transferidas do segundo pai porque já estavam presentes no filho.

No Passo 2 ainda da parte b) transferiu-se na mesma ordem, para as posições com \*, as cidades interiores aos pontos de corte do Pai 2 ausentes no Filho 1. Este mesmo processo deve ser executado para a construção do Filho 2 da parte c) da

Figura 13, entretanto são considerados o trecho interior aos pontos de corte no Pai 2 e as posições exteriores aos pontos de corte no Pai 1.

Variações deste operador podem ser construídas ao modificar os trechos aleatórios preservados nos filhos, por exemplo, com dois trechos temos o operador PMX2, com 4 o PMX4, e assim sucessivamente.

### 3.5.6.2 OPERADOR OX

O operador OX valoriza o subconjunto de cidades que ocupam o mesmo subconjunto de posições em ambos os pais. Para gerar o primeiro filho essas cidades são copiadas do primeiro pai (nas mesmas posições), e as outras posições são preenchidas com as cidades do segundo pai.

A posição de cada cidade é herdada de um dos dois pais. No entanto, muitas arestas podem ser quebradas no processo, porque o subconjunto inicial de cidades não está localizado necessariamente em posições consecutivas (POTVIN, 1996).

Na Figura 14, o grupo de cidades {3,4,6} ocupam as mesmas posições {2,4,5} nos cromossomos pais, não necessariamente na mesma ordem. O Filho 1 é criado copiando estas cidades do Pai 1. No passo 2, as outras posições marcadas com o símbolo \* recebem as cidades do Pai 2 que ocupam estas posições. Para gerar o Filho 2 procede-se analogamente tomando o grupo de cidades {3,4,6} como aparecem em Pai 2 e as outras posições marcadas com \* recebem as cidades de Pai 1.

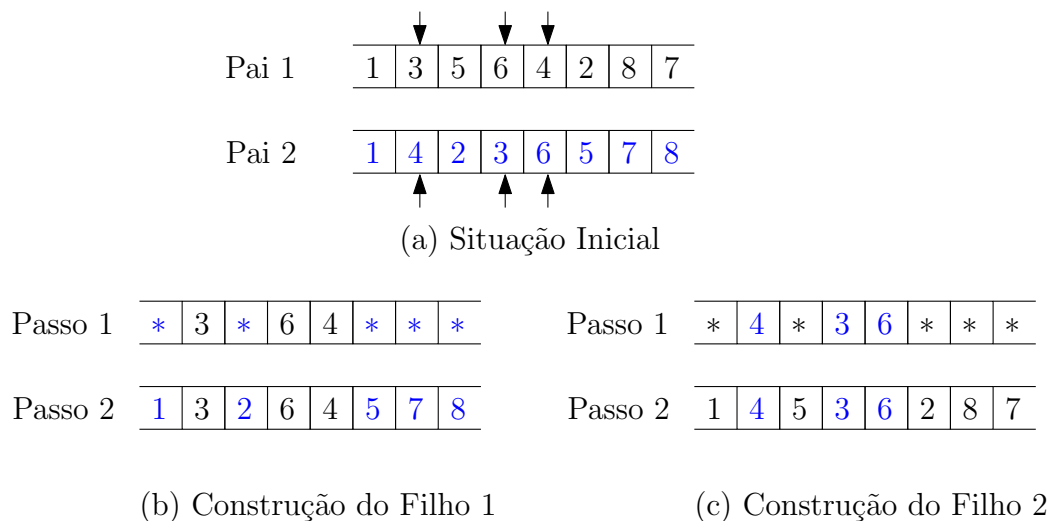


FIGURA 14: Construção de Descendentes com Operador OX

FONTE: Autor(2015)

### 3.5.6.3 OPERADOR CX

No operador CX, uma cidade só pode ocupar determinada posição no filho se ela estiver presente em um dos pais nesta mesma posição. Para satisfazer esta exigência e garantir que os filhos gerados sejam viáveis é preciso construir dois ciclos. Os ciclos indicam as posições que cada pai irá transferir para os filhos. Para formar o primeiro ciclo realiza-se uma sequência de movimentos que iniciam na primeira cidade de um dos pais e alterna entre um pai e outro até reencontrá-la no outro pai, seguindo as regras:

- Para migrar do primeiro até o segundo pai, desloca-se da posição atual para a posição correspondente no segundo pai (Figura 15(a)).
- Para migrar do segundo até o primeiro pai, procura-se a posição no primeiro pai que contém a cidade da posição atual no segundo pai (Figura 15(b)).

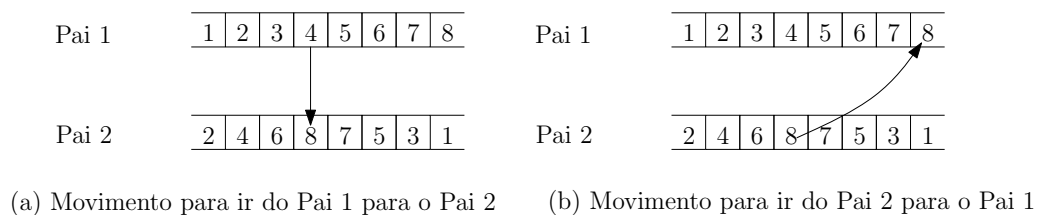


FIGURA 15: Movimentos Básicos Entre os Pais

FONTE: Autor(2015)

O segundo ciclo é formado pelas posições que não estão no primeiro ciclo. As posições visitadas no primeiro ciclo indicam as cidades que o primeiro pai transfere para o Filho 1 e as posições do segundo ciclo indicam as que o segundo pai fornece. Para gerar o Filho 2 basta tomar as cidades das posições do primeiro ciclo no segundo pai e as do segundo ciclo no primeiro pai.

Na Figura 16 temos um exemplo que ilustra a construção dos ciclos. Neste caso, o primeiro e o segundo ciclos são compostos pelas posições  $\{1, 2, 4, 8\}$  e  $\{3, 5, 6, 7\}$ , respectivamente. Estas posições serão utilizadas para construir os filhos na Figura 17.

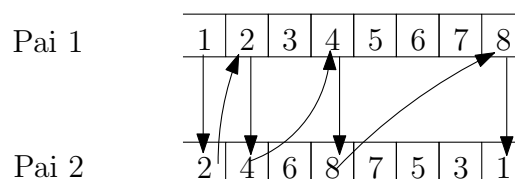


FIGURA 16: Construção dos Ciclos para Auxiliar o Operador CX

FONTE: Autor(2015)

As cores da Figura 17 sintetizam a ação dos ciclos, ou seja, para formar o Filho 1 associou-se as cidades em azul no primeiro pai (correspondentes as posições do primeiro ciclo) com as cidades em vermelho do segundo pai (correspondentes as posições do segundo ciclo). Para formar o Filho 2, juntou-se as cidades em preto (correspondentes as posições do primeiro ciclo) do primeiro pai com as cidades em verde do segundo pai (correspondentes as posições do segundo ciclo).

Pai 1	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr></table>	1	2	3	4	5	6	7	8	Filho 1	<table><tr><td>1</td><td>2</td><td>6</td><td>4</td><td>7</td><td>5</td><td>3</td><td>8</td></tr></table>	1	2	6	4	7	5	3	8
1	2	3	4	5	6	7	8												
1	2	6	4	7	5	3	8												
Pai 2	<table><tr><td>2</td><td>4</td><td>6</td><td>8</td><td>7</td><td>5</td><td>3</td><td>1</td></tr></table>	2	4	6	8	7	5	3	1	Filho 2	<table><tr><td>2</td><td>4</td><td>3</td><td>8</td><td>5</td><td>6</td><td>7</td><td>1</td></tr></table>	2	4	3	8	5	6	7	1
2	4	6	8	7	5	3	1												
2	4	3	8	5	6	7	1												

FIGURA 17: Construção de Descendentes com o Operador CX

FONTE: Autor(2015)

#### 3.5.6.4 OPERADOR MPX

Este operador atua de forma semelhante ao PMX. Primeiro seleciona-se um trecho aleatório no primeiro pai. Na sequência todas as cidades do trecho selecionado são removidas do segundo pai. Finalmente, o trecho selecionado no primeiro pai e a sobra do segundo pai após a eliminação são dispostos lado a lado para formar o filho (LARRANAGA *et al.*, 1999). A Figura 18 representa o operador MPX sendo aplicado sobre Pai 1 e Pai 2 para gerar o Filho.

Pai 1	1	2	3	4	5	6	7	8
Pai 2	2	4	6	8	7	5	3	1
				↓				
Filho	3	4	5	2	6	8	7	1

FIGURA 18: Construção de Descendentes com o Operador MPX

FONTE: Autor(2015)

No exemplo da Figura 18, os indivíduos possuem poucas cidades e a exigência do trecho ter mais que 10 cidades foi flexibilizada. A vantagem deste operador é que ele destrói um número limitado de arestas cada vez que é aplicado.

#### 3.5.6.5 OPERADOR ERX

Este operador é adequado para o PCV simétrico. Ele constrói os descendentes assumindo que somente o valor das arestas são importantes, e não suas orien-



tações. De acordo com esta suposição, as arestas são vistas como condutoras de informação genética. O operador ERX tenta passar o máximo de arestas dos pais para os filhos. A quebra destas é vista como uma mutação indesejável.

Um problema que ocorre em operadores com estratégias de recombinação de arestas é que eles costumam deixar arestas isoladas sem continuidade sendo necessário introduzir novas arestas para ligá-las. O ERX tenta sanar este problema escolhendo primeiro as cidades com menos opções de arestas para ligar-se.

Para ser executado, inicialmente cria-se uma lista de adjacências, contendo para cada cidade, suas duas vizinhas provenientes do primeiro pai e suas duas vizinhas provenientes do segundo pai, excluindo-se possíveis repetições. Na Figura 19 são mostrados dois pais e a lista de adjacências construída a partir deles.

	Cidade	Lista
Pai 1	1	2 6 3 5
	2	1 3 4 6
	3	2 4 1
Pai 2	4	3 5 2
	5	4 6 1
	6	1 5 2

FIGURA 19: Construção da Lista para Aplicar o Operador ERX

FONTE: Autor(2015)

Após a construção da lista, o filho é gerado seguindo o pseudocódigo:

ALGORITMO	
Passo 1:	Escolha uma cidade inicial em um dos dois pais (randomicamente ou por algum critério). Esta cidade ganha o <i>status</i> de “cidade atual”
Passo 2:	Remova todas as ocorrências da “cidade atual” do lado esquerdo da lista. Para evitar procuras extensas pode-se usar a coleção de arestas na linha da “cidade atual”.
Passo 3:	Se a linha da “cidade atual” na lista não estiver vazia, vá para o Passo 4, caso contrário vá para o Passo 5.
Passo 4:	Determine na linha da “cidade atual” qual a cidade que possui a menor quantidade de arestas em sua própria coleção de arestas. Faça desta cidade a “cidade atual” e vá para o Passo 2.
Passo 5:	Se a lista estiver toda vazia pare. Caso contrário, escolha randomicamente uma entre as cidades ainda não visitadas e vá para o Passo 2.

FONTE: Adaptado de Larranaga *et al.* (1999)

#### 3.5.6.6 OPERADOR CNP4

O operador CNP4, representa uma das inovações desta tese. Neste operador procura-se manter a ordem e a posição das cidades nos trechos transferidos para os

filhos, contudo, os pontos de corte não são os mesmos para ambos os pais, diferentemente dos operadores clássicos da literatura. O pseudocódigo do operador CNP4 é dado por:

---

**ALGORITMO: CNP4**

---

Dados de Entrada: **Pai 1, Pai 2**

Passo 1: Selecionar aleatoriamente dois trechos no primeiro pai e dois no segundo, sem haver sobreposições entre quaisquer dos 4 trechos.

Passo 2: Criar dois filhos contendo os trechos selecionados no passo anterior, mantendo as demais posições vazias.

Passo 3: Reparar os trechos transferidos para os filhos: Se nos filhos, alguma cidade se repetir, manter a primeira vez que aparece e excluir a outra.

Passo 4: Completar no Filho 1, as cidades ausentes, com cidades provenientes de Pai 2. Para isso manter a ordem e desprezar a posição.

Passo 5: Repetir o Passo 4 para Filho 2 e Pai 1.

Dados de Saída: **Filho 1 e Filho 2**

---

FONTE: Autor(2015)

Para exemplificar, consideremos duas soluções para um problema com 25 cidades (Pai 1 e Pai 2), apresentadas na Figura 20. Para estas cidades executou-se o Passo 1 do pseudocódigo do CNP4, selecionando aleatoriamente 4 trechos para serem transferidos para os filhos.

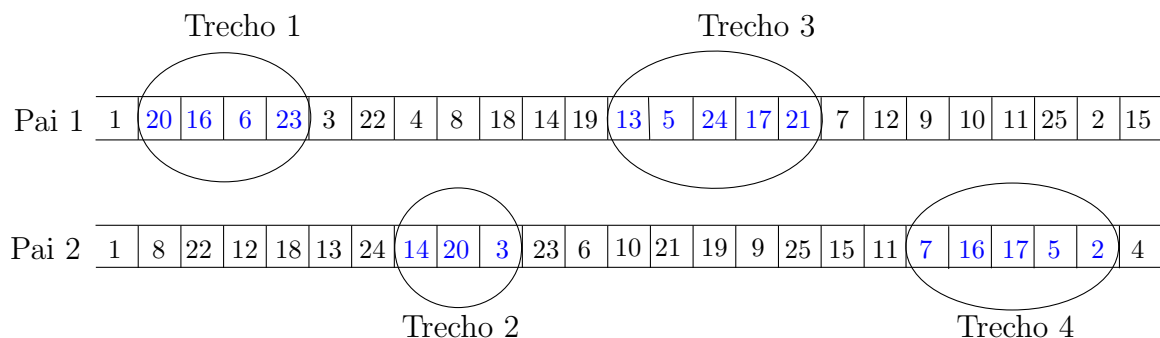


FIGURA 20: Resultado do Passo 1 do Operador CNP4

FONTE: Autor(2015)

A Figura 21 apresenta o Passo 2 do operador CNP4. Os trechos do Passo 1 são transferidos para os novos filhos (Filho 1 e Filho 2), sendo que o Pai 1 contribuiu com os trechos 1 e 3 enquanto que o Pai 2, com os trechos 2 e 4.

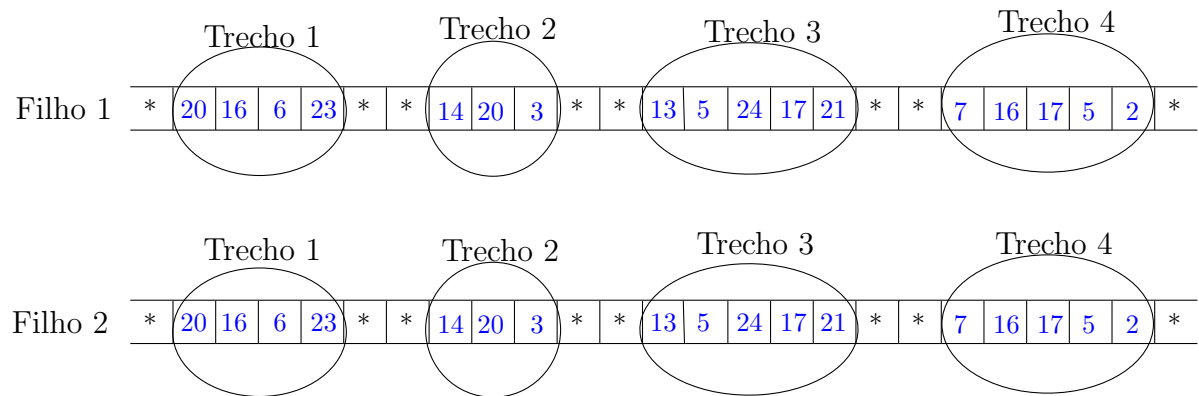


FIGURA 21: Resultado do Passo 2 do Operador CNP4

FONTE: Autor(2015)

No Passo 3 do algoritmo (Figura 22), são realizadas as reparações nos trechos recebidos dos pais, podendo ser observado que as cidades 20 (no Trecho 2), 16, 17 e 5 (no Trecho 4) são descartadas dos filhos por aparecerem em duplicidade após a junção dos trechos.

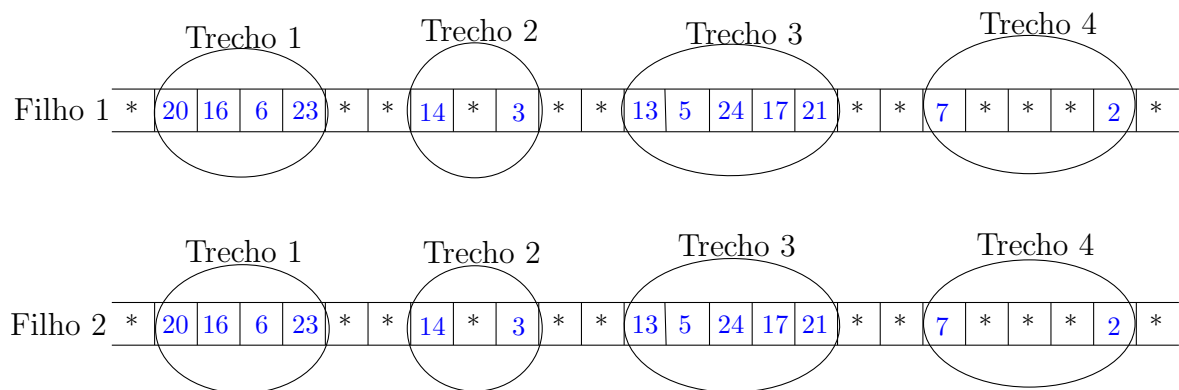


FIGURA 22: Resultado do Passo 3 do Operador CNP4

FONTE: Autor(2015)

Mostra-se na Figura 23, o Passo 4, onde são transferidas do Pai 2, todas as cidades ainda ausentes em Filho 1. A transferência segue a ordem em que estas cidades aparecem em Pai 2, contudo, são alocadas nas posições ainda disponíveis em Filho 1. As cidades transferidas estão em preto na figura.

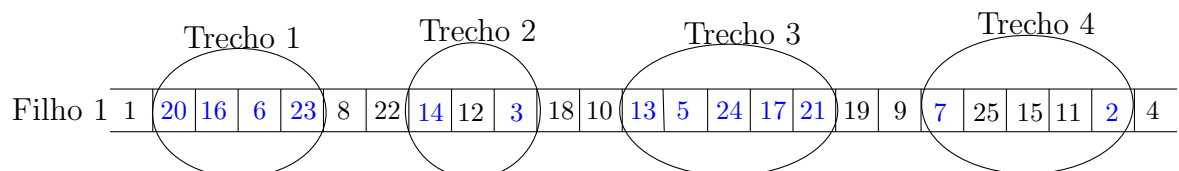


FIGURA 23: Resultado do Passo 4 do Operador CNP4

FONTE: Autor(2015)

No Passo 5, repete-se a ação do Passo 4 considerando o Filho 2 e o Pai 1. Novamente as cidades em preto mostradas na Figura 24 refletem o resultado da

execução deste passo.

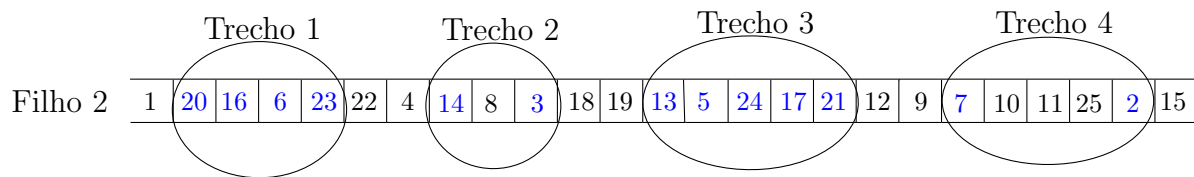


FIGURA 24: Resultado do Passo 5 do Operador CNP4

FONTE: Autor(2015)

### 3.5.6.7 OPERADOR MPH

O operador de Máxima Preservação Heurística (MPH) representa outra inovação neste trabalho. Este operador gera um único filho ao ser aplicado. O principal objetivo é contribuir para com o ajuste fino das soluções, ou seja, depois de inúmeras gerações a população irá dispor de alguns indivíduos muito evoluídos e, para estes casos, o operador é eficaz combinando-os para gerar um indivíduo ainda mais apto.

No pseudocódigo a seguir está descrito o operador MPH. Este operador difere-se do CNP4 por selecionar os trechos dos pais e manter no filho exatamente os trechos com duas ou mais arestas comuns aos pais.

---

#### ALGORITMO: MPH

---

Dados de Entrada: **População**

Criar **Lista** com indivíduos com os melhores *fitness*

Enquanto **Lista** de Indivíduos não estiver vazia. Faça

- Passo 1:    a) Selecionar aleatoriamente dois pais da **Lista**
- b) Reservar os trechos contendo duas ou mais arestas comuns entre os pais.
- Passo 2:        Construir subproblema e obter uma solução inicial para ele.
- Passo 3:        Aplicar heurística 3-Opt sobre a solução inicial do subproblema.
- Passo 4:    a) Reconstruir problema Original.
- b) Excluir pais atuais da **Lista**.

Fim

Dados de Saída: **Filhos** resultantes

---

FONTE: Autor(2015)

Para fomentar o entendimento, o operador MPH será aplicado em um exemplo fictício com 25 cidades. As duas soluções pais e os trechos do filho, determinados no Passo 1 do pseudocódigo do MPH, são apresentados na Figura 25.

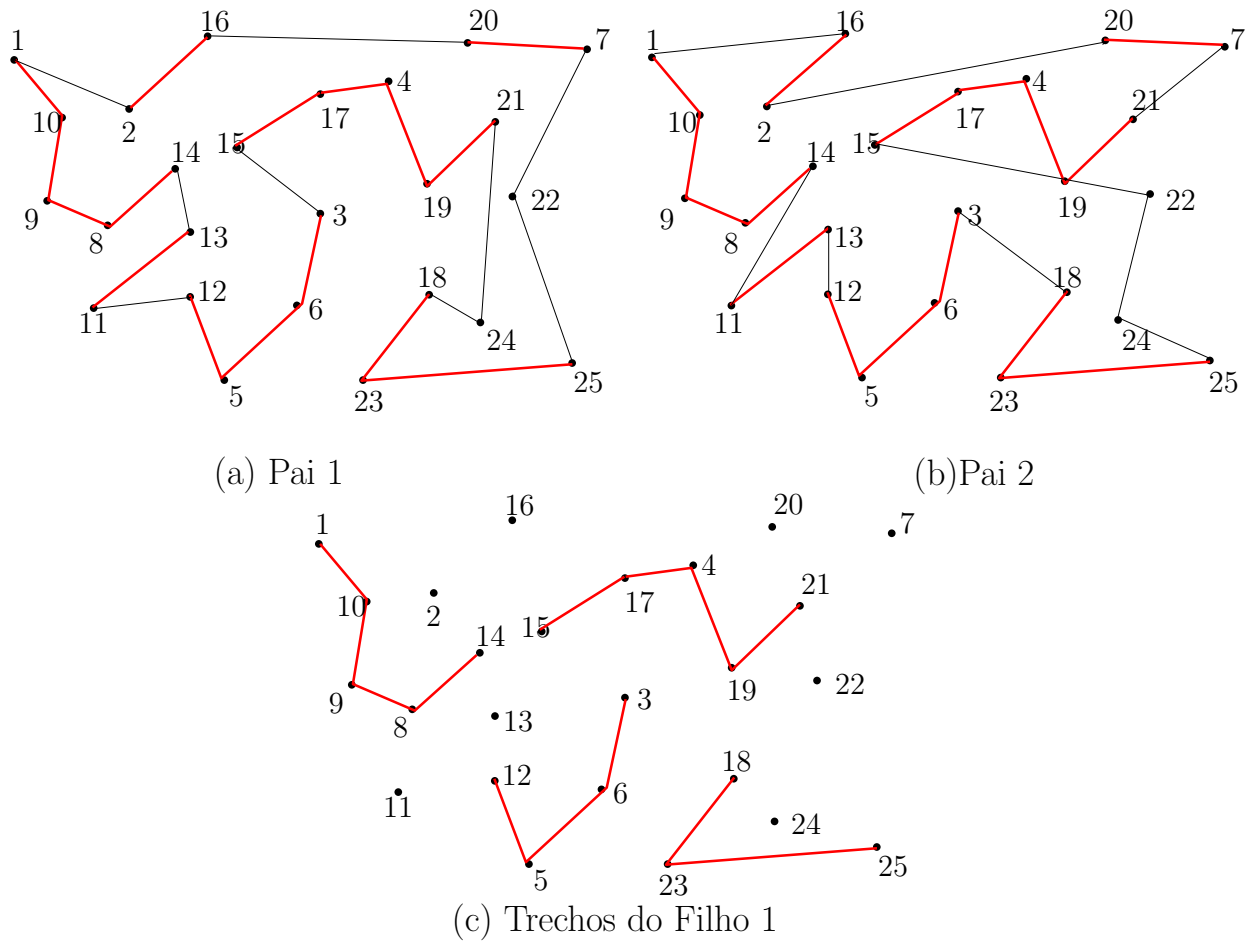
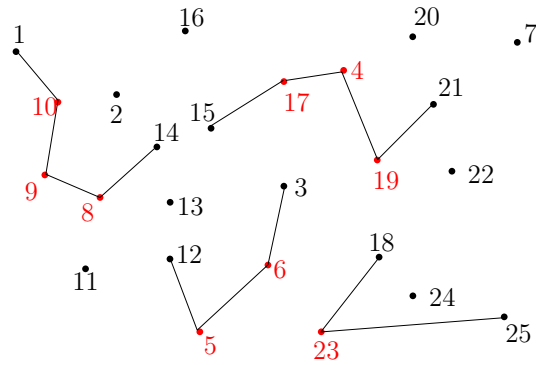


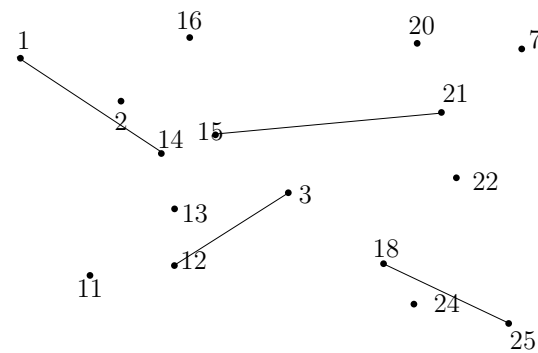
FIGURA 25: Resultado do Passo 1 do Operador MPH

FONTE: Autor(2015)

Para construir o subproblema mostrado na Figura 26(b), de que trata o Passo 2 do pseudocódigo do MPH, exclui-se todas as cidades, interiores aos trechos que serão transferidos para o filho (cidades em vermelho na Figura 26(a)). Após as exclusões, um subproblema com menos cidades e com a obrigatoriedade de ligação entre as cidades extremidades dos segmentos transferidos para o filho é obtido e está ilustrado na Figura 26(b).



a) Cidades do problema original e trechos a serem transferidos para o filho



b) Subproblema Gerado Após Exclusões

FIGURA 26: Subproblema do Passo 2 do Operador MPH

FONTE: Autor(2015)

Na sequência termina-se o Passo 2 e executa-se o Passo 3, gerando-se uma solução inicial para o subproblema, por qualquer método construtivo, e aplica-se a heurística 3-Opt para obter a solução apresentada na Figura 27.

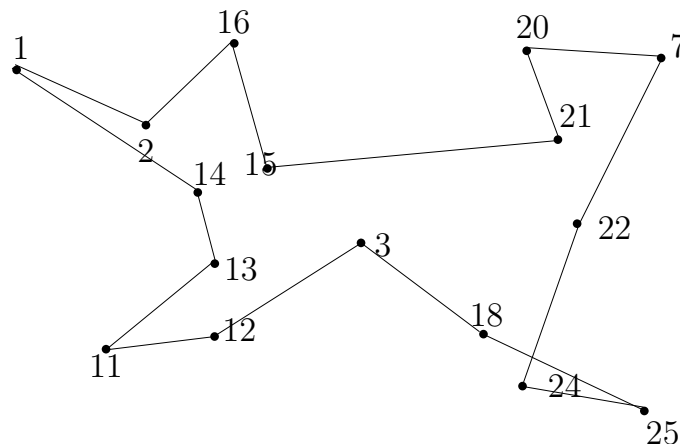


FIGURA 27: Solução Para o Subproblema

FONTE: Autor(2015)

Por fim, no Passo 4 faz-se a reinserção das cidades excluídas, entre as cidades extremidades dos segmentos da Figura 25(a) e tem-se uma solução filho (Figura

28) para o problema original.

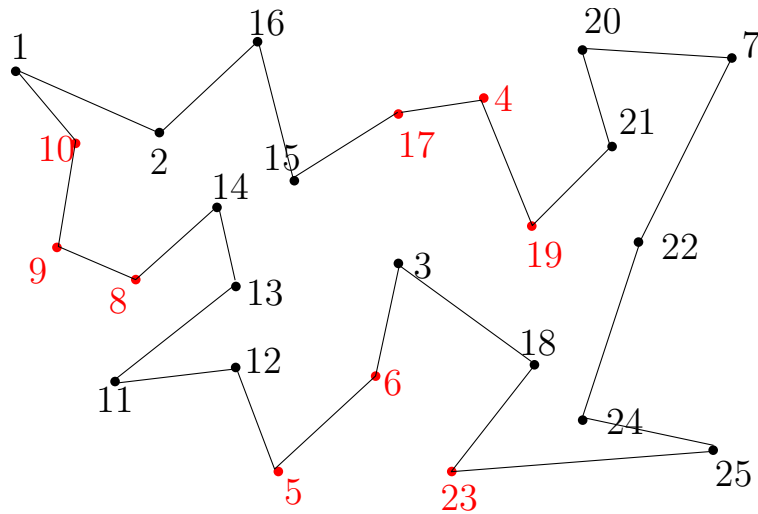


FIGURA 28: Filho Gerado pelo Operador MPH

FONTE: Autor(2015)

#### 3.5.6.8 OPERADOR MPHb

Assim como no operador MPH, este é outro operador proposto nesta tese, cujo objetivo é contribuir para o ajuste fino das soluções. O operador de Máxima Preservação Heurística - Versão b (MPHb) preserva somente um percentual das arestas comuns entre dois indivíduos. À seguir apresenta-se seu pseudocódigo.

---

##### ALGORITMO: MPHb

---

Dados de Entrada: **População** e **ParâmetroK**

Criar **Lista** com indivíduos com os melhores *fitness*

Enquanto **Lista** de Indivíduos não estiver vazia. Faça

Passo 1: a) Selecionar aleatoriamente dois pais da **Lista**

b) Reservar os trechos comuns entre os pais para o filho.

Passo 2: Desprezar aleatoriamente **ParâmetroK** arestas reservadas no Passo 1 (b)

Passo 3: Construir subproblema e obter solução inicial para ele.

Passo 4: Aplicar heurística 3-Opt sobre solução inicial do Passo 3.

Passo 5: a) Reconstruir problema Original.

c) Exclui pais atuais da **Lista**.

Fim

Dados de Saída: **Filhos** resultantes

---

FONTE: Autor(2015)

A diferença entre o operador para o MPH e o MPHb consiste somente no Passo 2, foi criado para o final do processo do AG. Após diversas gerações os melhores indivíduos são muito aptos e possuem diversas arestas comuns, em sua maioria coincidentes com as arestas da solução ótima. Entretanto, algumas delas apesar de

compartilhadas entre eles, não estão presentes na solução ótima. A estratégia é eliminar do filho algumas arestas, aleatoriamente, ou seja, desligar ao acaso uma parcela das arestas comuns aos pais e que deveriam ser transferidas para o filho se fossem seguidas as regras do operador, na expectativa de ter feito uma boa escolha e, assim, proporcionar ao filho que está sendo formado, uma maior chance de não conter arestas ausentes na solução ótima.

### 3.5.7 OPERADORES GENÉTICOS - MUTAÇÃO

Estes operadores são agentes de evolução secundários nos AGs, distinguindo-se dos operadores de cruzamento por serem aplicados sobre um único indivíduo e realizarem suaves modificações estruturais.

Na natureza, também ocorre algo parecido, ou seja, as mutações genéticas aparecem com baixa frequência sobre algum indivíduo e, quando se manifestam, modificam de forma mínima as características do indivíduo.

A função básica deste operador é agregar variabilidade genética à população, principalmente após diversas gerações de execução. Alguns problemas podem ser ocasionados em função do seu uso exagerado, como por exemplo, o retrocesso na aptidão adquirida. Ao longo dessas quase 4 décadas de estudo dos AGs, diversos operadores de mutação para o PCV foram apresentados na literatura. Alguns destes podem ser encontrados em Larranaga *et al.* (1999) e serão descritos nas próximas subseções.

#### 3.5.7.1 MUTAÇÃO POR DESLOCAMENTO

Este operador foi proposto por Michalewicz (1992). Para aplicá-lo seleciona-se um trecho do roteiro de forma aleatória. Este trecho é removido de sua região original inserido em outra região definida também aleatoriamente. A Figura 29 apresenta um exemplo.

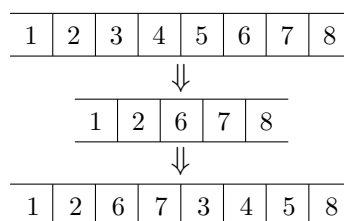


FIGURA 29: Exemplo de Mutação por Deslocamento

FONTE: Autor(2015)



O trecho escolhido foi {3,4,5}. Retira-o e, a seguir, seleciona-se um ponto para sua reinserção.

### 3.5.7.2 MUTAÇÃO POR TROCA

Para realizar a mutação por troca (BANZHAF, 1990), basta selecionar aleatoriamente duas cidades e trocá-las de posição no roteiro. A Figura 30 apresenta um exemplo destacando o roteiro antes e depois de aplicar o operador sobre as cidades 2 e 5.

1	2	3	4	5	6	7	8
↓							
1	5	3	4	2	6	7	8

FIGURA 30: Exemplo de Mutação por Troca

FONTE: Autor(2015)

### 3.5.7.3 MUTAÇÃO POR INSERÇÃO

Esta mutação se deve a Michalewicz (1992) e seu funcionamento consiste em escolher aleatoriamente uma cidade, removê-la do roteiro e inserir novamente em outra posição também aleatória. A Figura 31 apresenta dois roteiros que representam a mutação de inserção sendo aplicada ao retirar a cidade 4 de sua posição original e inserindo após a cidade 6.

1	2	3	4	5	6	7	8
↓							
1	2	3	5	6	4	7	8

FIGURA 31: Exemplo de Mutação por Inserção

FONTE: Autor(2015)

### 3.5.7.4 MUTAÇÃO POR SIMPLES INVERSÃO

A mutação por simples inversão, apresentada por Holland (1975), consiste em selecionar aleatoriamente dois pontos de corte no roteiro e na sequência inverter as posições de todas as cidades no trecho entre os pontos de corte.

De forma simplificada o operador de mutação por simples inversão pode ser entendido como um passo da heurística 2-Opt desenvolvida por Lin (1965). Normalmente este operador é usado em aplicações de *Simulated Annealing*.

Um exemplo de mutação por simples inversão pode ser observado na Figura 32 onde o trecho (345), que foi escolhido aleatoriamente, sofreu inversão, conforme se evidencia nos dois roteiros (antes e depois de aplicar a mutação).

1	2	3	4	5	6	7	8
⇓							
1	2	5	4	3	6	7	8

FIGURA 32: Exemplo de Mutação por Simples Inversão

FONTE: Autor(2015)

### 3.5.7.5 MUTAÇÃO POR INVERSÃO

O operador de mutação por inversão (FOGEL, 1988) é semelhantemente ao operador de mutação por deslocamento. Também ocorre a seleção aleatória e remoção de um trecho do roteiro, na sequência o trecho é inserido aleatoriamente e de forma reversa a partir de alguma posição do subroteiro resultante.

Na Figura 33 temos um exemplo da mutação por inversão para o trecho (345) reinserido de forma aleatória após a cidade 7. A figura apresenta o roteiro inicial, o subroteiro e o roteiro resultante.

1	2	3	4	5	6	7	8
⇓							
1	2	6	7	8			
⇓							
1	2	6	7	5	4	3	8

FIGURA 33: Exemplo de Mutação por Inversão

FONTE: Autor(2015)

### 3.5.8 ELITISMO

Ao final de uma iteração do AG, ou seja, após submeter os indivíduos da população corrente aos operadores de cruzamento e mutação, é necessário definir a maneira como ocorrerão as substituições destes indivíduos por seus descendentes.

São mencionadas na literatura pelo menos três formas de substituição: a substituição tradicional (AG Geracional), a substituição por estado estacionário (AG *Steady-State*) e a substituição por estado estacionário sem duplicação (AG *Steady-State* com elitismo) (ICHIHARA, 2013).

Na substituição tradicional, todos os indivíduos são trocados por seus descendentes, ao passo que na substituição por estado estacionário a troca é gradativa

e a cada iteração uma parcela dos indivíduos são escolhidos para dar lugar a seus descendentes. Por fim a substituição por estado estacionário sem duplicação é quase idêntica a forma anterior, com o acréscimo de impedir a entrada na população de um descendente que é réplica de algum pai.

No AG Geracional pode ocorrer ainda que o melhor indivíduo seja perdido no decorrer das gerações e, para este caso, Kenneth (1975) criou o operador de elitismo cujo objetivo é obrigar que os melhores indivíduos da geração corrente, sejam incluídos na próxima geração.

### 3.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo foi descrito o PCV e foram apresentadas aplicações reais para motivar o estudo do problema. Também descreveu-se algumas das metodologias presentes na literatura, tanto relacionadas a modelagem exata do problema quanto às heurísticas. Para este segundo grupo deu-se atenção especial aos AGs e todos os seus detalhes de funcionamento, tais como: métodos de seleção para cruzamento, operadores genéticos disponíveis e seleção de indivíduos para as gerações futuras.

A atenção especial aos AGs deve-se à implementação que é apresentada no próximo capítulo, que propõe um AG híbrido para resolver o PCV. A população inicial foi construída com uma adaptação da heurística do vizinho mais próximo, detalhada anteriormente. O AG clássico recebeu diversas modificações, entre elas, a incorporação da heurística de melhoria 3-Opt, o que lhe conferiu o caráter de método híbrido.

## 4 IMPLEMENTAÇÃO COMPUTACIONAL

Neste capítulo relata-se a implementação computacional, tanto das versões adaptadas que envolvem modificações nas regras internas de funcionamento de um AG clássico, quanto a versão adaptada e híbrida. As modificações foram implementadas em AGs adaptados (versões AG Acs, AG Padronizar, AG Rinap, AG Trechos contendo, cada versão, a modificação em uma regra específica e a versão AGA, contendo as quatro modificações simultaneamente). A versão adaptada e híbrida (AGAH) envolve um AG que, além das adaptações, associou-se a heurística 3-Opt.

### 4.1 DESCRIÇÃO DO AGA PROPOSTO E SUAS VARIAÇÕES

Entre os objetivos específicos do trabalho está revisar as regras internas de funcionamento de um AG clássico para torná-lo mais eficaz. O que motivou a elaboração do AGA, que agrega quatro modificações nas regras internas de funcionamento do AG clássico. A descrição das implementações das variações do AGA (AGAcS, AGPadronizar, AGRinap, AGTrechos) será omitida porque seguem a implementação proposta para ele, com exceção de que em cada versão foi realizada uma adaptação por vez. Para o entendimento do AGA, apresenta-se o seu pseudocódigo a seguir.

---

#### ALGORITMO: AGA

---

Dados de Entrada: Matriz de **Distâncias** e os Parâmetros do AGA

GerarPopulação Inicial

Enquanto **Iteração** for menor que **CritérioParada**. Faça

    Se o critério para execução for satisfeito, Executar Rinap

    Avaliar *Fitness* da **População**

    Parar se a melhor solução conhecida na literatura estiver na População

    Determinar **Lista** de Indivíduos para Cruzamento

    Enquanto percorre a **Lista** de Cruzamentos. Faça

        Realizar o Cruzamento em um Casal da Lista com o operador CNP4

        Realizar a mutação nos filhos

        Determinar 2 sobreviventes: entre o casal e seus dois filhos

    Fim

    Realizar Elitismo

    Padronizar a **População**

Fim

Dados de Saída: **População**

---

FONTE: Autor(2015)

O fluxograma da Figura 34 apresenta outra forma de descrever o funcionamento do AGA.

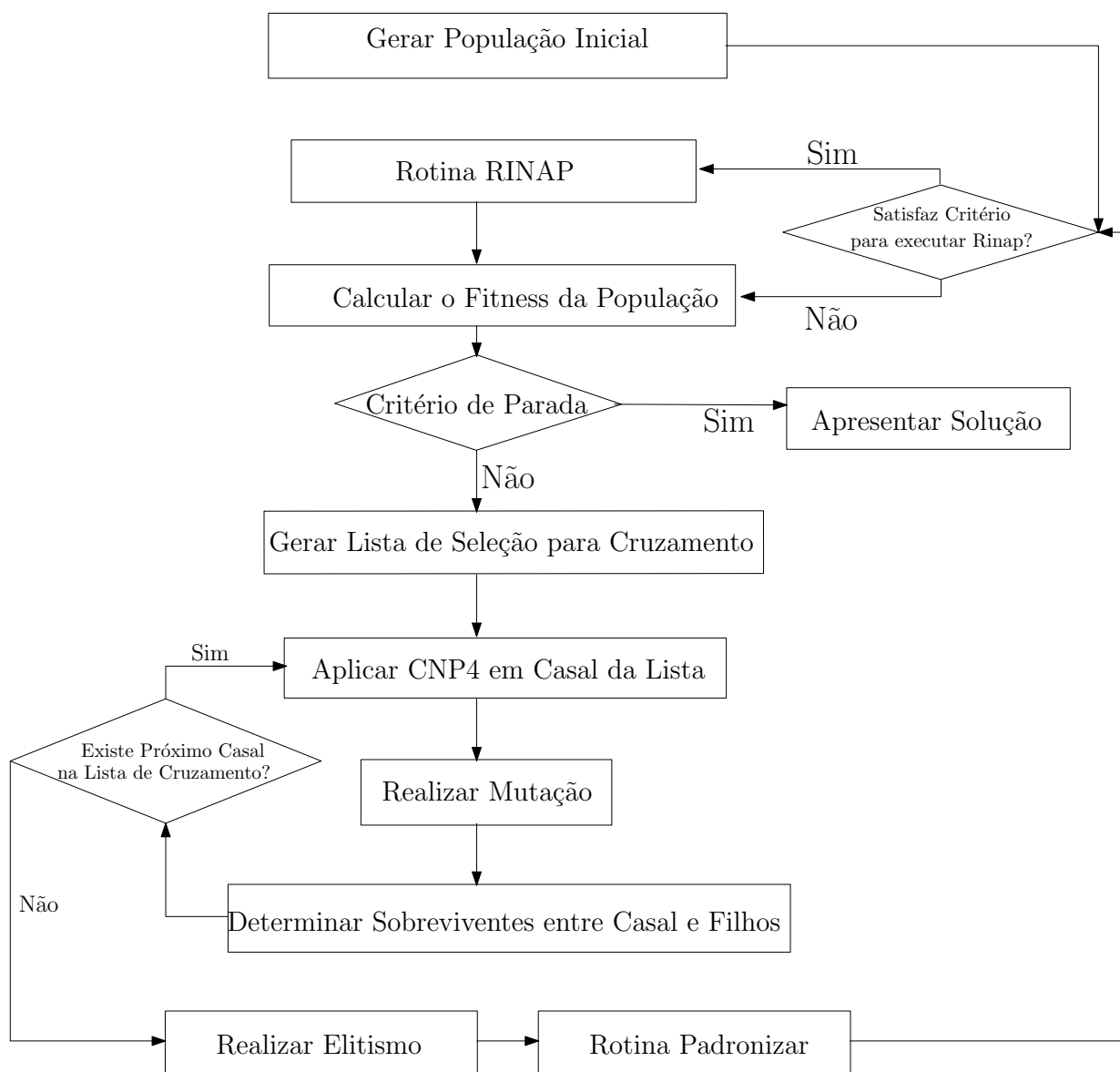


FIGURA 34: Fluxograma do AGA

FONTE: Autor(2015)

#### 4.1.1 BANCO DE DADOS PARA OS TESTES

Para avaliar o desempenho do AGA foram utilizadas 40 instâncias disponíveis na biblioteca TSPLIB (REINELT, 1990), contendo até 400 cidades.

#### 4.1.2 INFORMAÇÕES DE *HARDWARE* E *SOFTWARE*

Os testes envolvendo o AGA ou uma de suas variações foram executados em dois computadores pessoais com processadores Intel(R) core(TM) i5 com CPUs de 3.20 GHz e 6 GB de RAM com Windows 8.1 Pro de 64-bits como sistema operacional. Também foi utilizado um *notebook* ACER com processador Intel(R) core(TM) i7 com CPU de 2.0 GHz e 16 GB de RAM equipado com o Windows 8.1 Pro de 64-bits como sistema operacional.

O código foi implementado em MatLab versão R2013b e para os testes estatísticos usou-se o software MINITAB Release 14.1.

#### 4.1.3 PARÂMETROS DO AGA

Diversos autores mencionam que uma boa escolha dos parâmetros é fundamental para obter resultados satisfatórios com a execução de um AG. Os principais parâmetros são a taxa de mutação, de cruzamento e o tamanho da população. Estes parâmetros definem, implicitamente, a velocidade com que a população evolui, a variabilidade genética presente e o quanto do espaço de busca estará representado.

Na Tabela 7 são apresentados cada um dos parâmetros do AGA implementado neste trabalho, uma breve descrição dos mesmos e o valor ou intervalo de valores que lhes foram atribuídos durante os testes.

TABELA 7: Parâmetros para o AGA proposto neste trabalho

Parâmetro	Valor ou Intervalo	Descrição
Tamanho	[17,400]	Tamanho dos problemas testados
TamPop	20 ou 100	Tamanho da população
PercMutação	10%	Percentual de mutação
PercCruzamento	70%	Percentual de Cruzamento
PercElitismo	1	Indivíduos preservados pelo elitismo
CritérioParada	10000	Quantidade de gerações permitidas
ParâmetroA	$\min\{0.10 \times \text{Tamanho}; 500\}$	Tamanho máximo dos trechos para o sorteio dos pontos de corte
ParâmetroB	2	Quantidade de vizinhos considerados ao gerar a população inicial
ParâmetroD	20	Quantidade de vezes que uma mesma instância é resolvida
ParâmetroE	40%	Percentual de genes de cada indivíduo a sofrer mutação
ParâmetroH	$0.80 \times \text{CritérioParada}$	Divisor entre gerações iniciais e finais
ParâmetroI	$\min\{0.10 \times \text{Tamanho}; 20\}$	Raio de vizinhança para construção do vetor Lista

continua para próxima página

continuação da página anterior		
Parâmetro	Valor ou Intervalo	Descrição
ParâmetroM	400	Intervalo de gerações para executar a rotina RINAP
ParâmetroN	5	Quantidade de novos indivíduos a serem inseridos na população pela rotina RINAP
LimiteA	0,30	Porcentagem de referência utilizada nas iterações iniciais da rotina Acs
LimiteB	0,60	Porcentagem de referência utilizada nas iterações intermediárias da rotina Acs
LimiteC	0,90	Porcentagem de referência utilizada nas iterações finais da rotina Acs

FONTE: Autor(2015)

#### 4.1.4 ESTRUTURA DE DADOS ADOTADA

Neste trabalho optou-se pela representação por caminho para o AGA, pois esta representação mostrou-se adequada, sendo amplamente escolhida entre os pesquisadores. É de fácil implementação, dispõe de diversos operadores implementados tanto de cruzamento quanto de mutação, e não é difícil corrigir indivíduos anômalos obtidos durante o processo.

#### 4.1.5 GERAÇÃO DA POPULAÇÃO INICIAL

Para a geração da população inicial foi utilizado o método descrito na subsubseção 3.5.4.1 com os parâmetros presentes na subseção 4.1.3.

#### 4.1.6 FUNÇÃO DE APTIDÃO DA POPULAÇÃO

Considerou-se como função de aptidão, o somatório das distâncias necessárias para se percorrer uma solução do PCV. Esta é a função clássica de aptidão encontrada na literatura.

#### 4.1.7 ROTINA RINAP

A rotina de Reinscrição de Indivíduos Novos e Aleatórios na População (RINAP) foi executada a cada intervalo de 400 gerações. Esta quantidade foi determinada empiricamente para que todos os indivíduos tenham tempo de contribuir com a evolução da população, visando aumentar a diversidade genética e evitar a convergência prematura.

Em uma analogia aos processos biológicos da natureza, seria o equivalente a admitir, em uma população isolada de indivíduos, a chegada de alguns indivíduos advindos de outras populações. Para cumprir este propósito, utiliza-se o algoritmo da seção 3.5.4.1 para obter “ParâmetroN” novos indivíduos e, na sequência, seleciona-se aleatoriamente a mesma quantidade de indivíduos da população para serem substituídos pelos novos. São inseridos poucos indivíduos para não causar perda excessiva de material genético acumulado ao longo das gerações.

Esta rotina é uma tentativa de simplificar a ação relatada em diversos trabalhos onde se usam populações evoluindo em paralelo e controladamente trocam indivíduos entre si (NGUYEN *et al.*, 2007b), (GHORPADE; KAMALAPUR, 2014) e (HONDA; NAGATA; ONO, 2013).

#### 4.1.8 MÉTODO DE SELEÇÃO PARA CRUZAMENTO E MUTAÇÃO

Para o algoritmo AGA adotou-se o procedimento que associa o método do Torneio à rotina ACS como método de seleção, discutidos na subseção 3.5.5.

#### 4.1.9 OPERADORES DE CRUZAMENTO

Utilizou-se o CNP4 como operador de cruzamento no AGA, este operador foi apresentado na subsubseção 3.5.6.6.

#### 4.1.10 OPERADORES DE MUTAÇÃO

Para o operador de mutação não foram propostas alterações em relação ao clássico. Foram implementados os operadores de mutação por troca e por inversão conforme, descrito no capítulo 3.

#### 4.1.11 CRITÉRIO PARA AVALIAR OS SOBREVIVENTES

Ao fim de cada geração, em um AG *Steady-State* com elitismo, determina-se entre pais e filhos, os sobreviventes para a geração futura. Neste trabalho, adaptou-se esta operação e, assim que concebido um novo filho, é avaliada sua inserção imediata na população. Devido a esta regra, pode ocorrer que um pai, selecionado para mais de um cruzamento na geração atual, seja substituído por um de seus filhos, restando ao filho que o substituiu, o papel de representá-lo nos próximos cruzamentos agendados para o pai.



Em resumo, o AGA foi implementado usando *Steady-State* com elitismo e a modificação extra de substituição imediatamente após o cruzamento.

#### 4.1.12 CRITÉRIO DE PADRONIZAÇÃO DA POPULAÇÃO

Este critério foi elaborado para evitar que o AGA avaliasse soluções desnecessárias. É possível que duas soluções contendo as cidades em posições diferentes representem o mesmo indivíduo por começarem em pontos distintos. Se a população for padronizada, o espaço de busca do problema é reduzido sem risco de perda da solução ótima.

Acredita-se que a padronização melhora a ação dos operadores de cruzamento por reduzir os reparos necessários durante a transferência de trechos dos pais para os filhos. A aplicação deste critério é um procedimento simples que consiste em obrigar que todas as soluções iniciem à partir da mesma cidade a cada geração.

Na Figura 35(a) tem-se uma população de soluções para um PCV com 11 cidades que não está padronizada, enquanto que na Figura 35(b), a mesma população é apresentada padronizada, ou seja, todos os indivíduos iniciam suas rotas de uma mesma cidade.

Indivíduo 1	2	8	5	10	3	1	7	4	6	11	9
Indivíduo 2	8	6	7	10	2	5	9	3	11	1	4
Indivíduo 3	4	2	8	1	10	5	7	9	6	11	3
Indivíduo 4	5	1	9	11	4	8	6	3	7	10	2
Indivíduo 5	7	6	8	5	10	3	2	11	9	4	1

(a) População não padronizada

Indivíduo 1	1	7	4	6	11	9	2	8	5	10	3
Indivíduo 2	1	4	8	6	7	10	2	5	9	3	11
Indivíduo 3	1	10	5	7	9	6	11	3	4	2	8
Indivíduo 4	1	9	11	4	8	6	3	7	10	2	5
Indivíduo 5	1	7	6	8	5	10	3	2	11	9	4

(b) População padronizada

FIGURA 35: Padronização da População

FONTE: Autor(2015)

## 4.2 DESCRIÇÃO DO AGAH PROPOSTO

O AGAH é uma versão híbrida do AGA, pois faz uso da heurística 3-Opt em sua execução. O AGAH também utiliza os operadores de cruzamento CNP4, MPH e o MPHb. Todas as modificações, bem como a motivação para a implementação dos novos operadores se deve a dois argumentos:

- Historicamente, os AGs não apresentam bom desempenho para o PCV, se comparados à heurísticas de busca local. Segundo Nguyen *et al.* (2007b), os AGs puros implementados em diversos trabalhos foram desencorajadores, tanto em relação ao tempo computacional quanto aos resultados obtidos com a utilização de heurísticas clássicas.
- Diversos pesquisadores têm concentrado esforços no estudo de AGs híbridos, na proposição de novos operadores, na melhoria das regras de funcionamento de um AG e na implementação de AGs com processamento em paralelo.

As etapas do AGAH presentes no Fluxograma da Figura 36 podem ser observadas também no pseudocódigo do algoritmo, apresentado a seguir.

---

#### ALGORITMO: AGAH

---

Dados de Entrada: Matriz de **Distâncias** e os Parâmetros do AGAH

Gerar População inicial

Aplicar 3-Opt sobre População inicial

Enquanto **Iteração** for menor que **CritérioParada**. Faça

Se o critério para execução for satisfeito, Executar Rinap

Avaliar *Fitness* da População

Parar se a melhor solução da literatura estiver na População

Determinar **Lista** de Indivíduos para Cruzamento usando Torneio+ACS

Enquanto percorre a **Lista** de Cruzamentos. Faça

Realizar o Cruzamento em um Casal da Lista com o operador CNP4

Realizar mutação sobre os filhos

Aplicar 3-Opt sobre os filhos

Determinar 2 sobreviventes: entre o casal e seus dois filhos

Fim

Determinar **ListaB** de Indivíduos para Cruzamento

Enquanto percorre a **ListaB** de Cruzamentos. Faça

Se **Iteração** for maior que **ParâmetroJ**

Realizar o Cruzamento em um Casal da Lista com o operador MPH

Fim

Se **Iteração** for maior que **ParâmetroM**

Realizar o Cruzamento em um Casal da Lista com o operador MPHb

Fim

Determinar 2 sobreviventes: entre o casal os 2 filhos, caso os filhos tenham sido gerados

Fim

Realizar Elitismo

Padronizar a População

Fim

Dados de Saída: **População**

---

Para melhor entendimento do funcionamento do AGAH, apresenta-se na Figura 36 um fluxograma contendo todas as suas etapas.

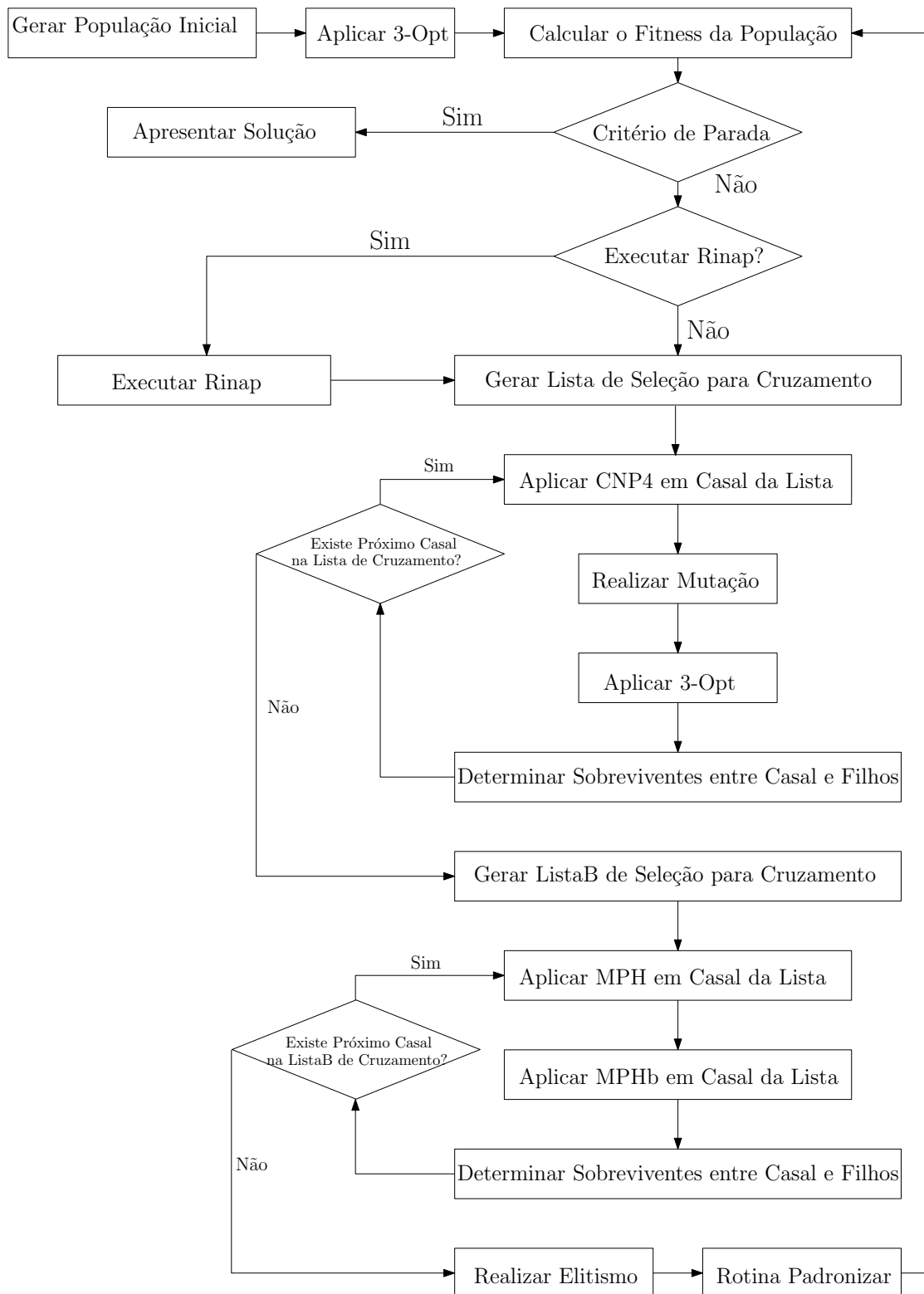


FIGURA 36: Fluxograma do AGAH

FONTE: Autor(2015)

#### 4.2.1 BANCO DE DADOS PARA OS TESTES

Para avaliar o AGAH foram utilizadas 85 das 111 instâncias disponíveis na biblioteca TSPLIB (REINELT, 1990). Elas formam uma coletânea com características variadas, contendo inclusive, dados relativos a aplicações reais. Foram desprezadas 16 instâncias por motivos variados, na Tabela 8 apresentam-se as instâncias desprezadas e o motivo de não terem sido utilizadas.

TABELA 8: Instâncias do TSPLIB não resolvidas pelo AGAH

Instância	Explicação
gr17	Não possui Coordenadas
gr21	Não possui Coordenadas
brg180	Dados Incompreensíveis
linhp318	Tempo de execução inviável
fl3795	Tempo de execução inviável
rl5915	Tempo de execução inviável
rl5934	Tempo de execução inviável
pla7397	Tempo de execução inviável
rl11849	Memória Insuficiente
usa13509	Memória Insuficiente
brd14051	Memória Insuficiente
fnl4461	Memória Insuficiente
d15112	Memória Insuficiente
d18512	Memória Insuficiente
pla33810	Memória Insuficiente
pla85900	Memória Insuficiente

FONTE: Autor(2015)

#### 4.2.2 INFORMAÇÕES DE *HARDWARE* E *SOFTWARE*

Nos testes envolvendo o AGAH foi utilizada a mesma configuração nos mesmos equipamentos e utilizados os mesmos *softwares* que no AGA.

#### 4.2.3 PARÂMETROS DO AGAH

Na Tabela 9 são apresentados cada um dos parâmetros necessários a execução do AGAH, uma breve descrição dos mesmos e o valor ou intervalo de valores que lhes foram atribuídos.

TABELA 9: Parâmetros para o AGAH proposto neste trabalho

Parâmetro	Valor ou Intervalo	Descrição
Tamanho	[17,3038]	Tamanho dos problemas testados
TamPop	25	Tamanho da população
PercMutação	10%	Percentual de mutação
PercCruzamento	70%	Percentual de Cruzamento
Elitismo	1	Indivíduos preservados pelo elitismo
CritérioParada	1000	Quantidade de gerações permitidas
ParâmetroA	$\min\{0.10 \times \text{Tamanho}; 500\}$	Tamanho máximo dos trechos para o sorteio dos pontos de corte
ParâmetroB	2	Quantidade de vizinhos considerados ao gerar a população inicial
ParâmetroC	10	Quantidade de cruzamento que se realiza com os operadores MPH e MPHb a cada geração que são acionados
ParâmetroD	10	Quantidade de vezes que uma mesma instância é resolvida
ParâmetroE	40%	Percentual de genes de cada indivíduo a sofrer mutação
ParâmetroH	$0.80 \times \text{CritérioParada}$	Divisor entre gerações iniciais e finais
ParâmetroI	$\min\{0.10 \times \text{Tamanho}; 20\}$	Raio de vizinhança para construção do vetor Lista
ParâmetroJ	$0.01 \times \text{CritérioParada}$	Determina a partir de qual geração o operador MPH é acionado
ParâmetroK	$\min\{0.10 \times \text{Tamanho}; 20\}$	Quantidade de arestas desprezadas pelo operador MPHb
ParâmetroL	$0.01 \times \text{CritérioParada}$	Geração para a qual o operador MPHb começa a ser acionado
ParâmetroM	400	Intervalo de gerações para executar a rotina RINAP
Faixa	10	Quantidade de vizinhos mais próximos considerados ao tentar as trocas 3-Opt
LimiteA	0,30	Porcentagem de referência utilizada nas iterações iniciais da rotina Acs
LimiteB	0,60	Porcentagem de referência utilizada nas iterações intermediárias da rotina Acs
LimiteC	0,90	Porcentagem de referência utilizada nas iterações finais da rotina Acs

FONTE: Autor(2015)

#### 4.2.4 ESTRUTURA DE DADOS ADOTADA

A estrutura de dados adotada para o AGAH foi a representação por caminho. Conforme mencionado anteriormente, Esta representação mostrou-se adequada, sendo amplamente escolhida entre os pesquisadores. É de fácil implementação, dispõe de diversos operadores implementados tanto de cruzamento quanto de mutação,

e não é difícil corrigir indivíduos anômalos obtidos durante o processo.

#### 4.2.5 GERAÇÃO DA POPULAÇÃO INICIAL

A população inicial também foi gerada a partir do método descrito na sub-subseção 3.5.4.1 com os parâmetros presentes na subseção 4.2.3. Contudo, assim que gerada a população inicial, aplicou-se a heurística 3-Opt sobre cada um de seus indivíduos, para então iniciar a execução do AGAH.

#### 4.2.6 FUNÇÃO DE APTIDÃO DA POPULAÇÃO

Considerou-se como função de aptidão, o somatório das distâncias necessárias para se percorrer uma solução do PCV. Esta é a função clássica de aptidão encontrada na literatura.

#### 4.2.7 ROTINA RINAP

A rotina RINAP foi implementada para o AGAH da mesma forma que no AGA.

#### 4.2.8 MÉTODO DE SELEÇÃO PARA CRUZAMENTO E MUTAÇÃO

Para o operador CNP4, o método de seleção dos indivíduos para cruzamento e mutação foi o mesmo adotado no AGA, ou seja, o método do Torneio e a rotina ACS. Para os operadores MPH e MPHb, a cada geração foram selecionados “ParâmetroC” duplas aleatoriamente entre os indivíduos mais aptos da população.

#### 4.2.9 OPERADORES DE CRUZAMENTO

O AGAH foi implementado com três operadores de cruzamento, executados em paralelo (CNP4, MPH e MPHb). Estes operadores foram descritos anteriormente.

#### 4.2.10 OPERADORES DE MUTAÇÃO

Aplicou-se os operadores de mutação por inversão e aleatória, somente sobre um percentual dos filhos gerados pelo operador de cruzamento CNP4. Este percentual foi determinado pelo parâmetro “PercMutaçao”, descrito anteriormente. O percentual dos genes modificados em cada filho selecionado para sofrer a ação dos operadores de mutação foi determinado por “ParâmetroE”.

#### 4.2.11 HEURÍSTICA 3-OPT

Neste trabalho a heurística 3-Opt foi adaptada, sendo proposto um critério para determinar a ordem em que as cidades são escolhidas para avaliar as possíveis trocas. Pressupõe-se que é mais urgente analisar as cidades que mais oneram a qualidade da solução, se não estiverem ligadas aos seus vizinhos mais próximos na solução final.

Sendo assim, utilizou-se a função dada pela Equação 9 para determinar a ordem de escolha das cidades por uma composição da média e o desvio padrão das distâncias de cada cidade até seus vizinhos mais próximos. A proximidade admitida foi limitada pelo parâmetro “Faixa”. A cidade cuja média é menor e possui menor desvio padrão tem também menor urgência de análise porque se ela não estiver ligada aos seus vizinhos mais próximos, o impacto sobre o custo final do trajeto será menor. Cidades com média alta e desvio padrão alto tem grande prioridade nas análises.

$$Score_i = Media_i + Desvio_i, \quad \forall i \in \{1, \dots, N\} \quad (9)$$

sendo que:

$$Media_i = \frac{\sum_{j=1}^{Faixa} Distancia_{ij}}{Faixa} \quad (10)$$

e ainda:

$$Desvio_i = \frac{\sum_{j=1}^{Faixa} (Distancia_{ij} - Media_i)^2}{Faixa - 1} \quad (11)$$

onde  $Distancia_{ij}$  representa distância de uma cidade  $i$  até o seu  $j$ -ésimo vizinho mais próximo.

Para exemplificar a aplicação do critério aqui proposto, consideremos o exemplo da Figura 37, no qual é apresentado a localização cartesiana de 6 cidades, um quadro contendo as distâncias médias de cada cidade até seus vizinhos mais próximos, o desvio padrão referente à essas distâncias, o score que representa a soma de ambos e a ordem. Esta última considera o score de forma decrescente para indicar qual a prioridade de cada cidade nas análises quando a heurística 3-Opt estiver em execução.

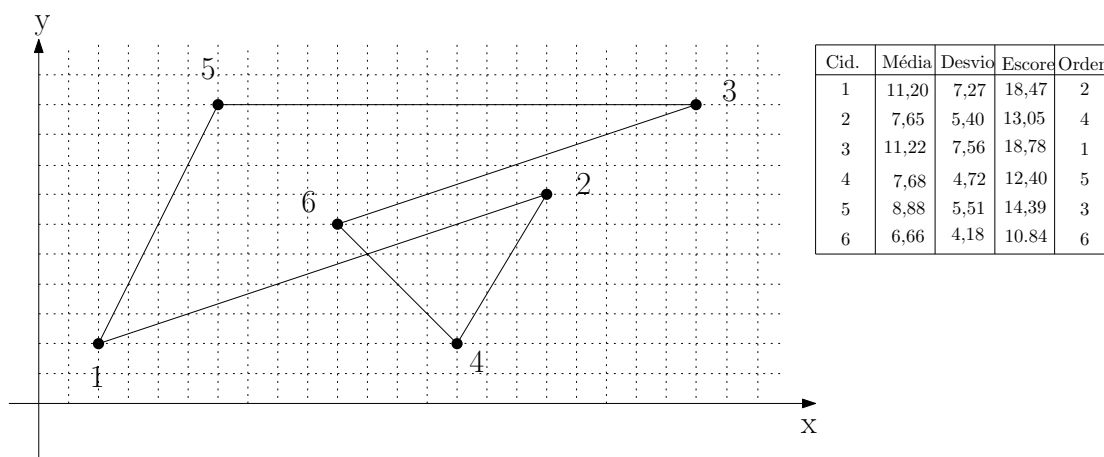


FIGURA 37: Exemplo para Aplicar o Critério Estatístico

FONTE: Autor(2015)

Para determinar as distâncias médias e os desvios, pode-se utilizar a matriz de distâncias apresentada na Tabela 10. Com base nestas informações e considerando o parâmetro “Faixa” igual a 5, pode-se determinar os valores presentes no quadro da Figura 37.

TABELA 10: Distancias entre as Cidades da Figura 37

Cidades	1	2	3	4	5	6
1	0	15,8114	21,5407	12	8,9443	8,9443
2	15,8114	0	5,831	5,831	11,4018	7,0711
3	21,5407	5,831	0	11,3137	16	12,6491
4	12	5,831	11,3137	0	11,3137	5,6569
5	8,9443	11,4018	16	11,3137	0	5,6569
6	8,9443	7,0711	12,6491	5,6569	5,6569	0

FONTE: Autor(2015)

No exemplo da Figura 37 a cidade 3 tem maior escore e foi atribuído a ela a ordem 1, ou seja, esta é a primeira cidade que a heurística 3-Opt deve considerar para realizar as trocas na tentativa de diminuir as distâncias entre seus dois vizinhos na solução final. Na solução atual esta cidade está conectada às cidades 6 e 5 mas seu vizinho mais próximo é a cidade 2. Se a cidade 3 for priorizada durante as melhorias, as chances de ela estar ligada à cidade 2 na solução final é maior. Graficamente esta ligação parece lucrativa porque a aresta (3,2) é muito menor que as arestas (3,5) e (3,6).

A cidade 6 é aquela que tem a menor prioridade pelo critério adotado neste trabalho, a explicação gráfica é a seguinte: caso esta cidade não esteja ligada aos seus vizinhos mais próximos que são 4 e 2, as opções restantes, por exemplo 1 ou 5, formam arestas não muito maiores que as arestas (6,2) ou (6,4).



A heurística 3-Opt, conforme descrita anteriormente, foi incorporada no AGAH em quatro rotinas, a primeira visando melhorar as soluções iniciais, na sequência sobre os filhos gerados pelo operador CNP4 e as outras duas como parte da aplicação dos operadores MPH e MPHb. Estas situações estão explícitas nos pseudocódigos e fluxogramas dos operadores CNP4, MPH e MPHb e dos AGs implementados.

#### 4.2.12 CRITÉRIO DE PADRONIZAÇÃO DA POPULAÇÃO

A padronização da população para o AGAH foi implementada da mesma forma que descrito na subsubseção 4.1.12.

## 5 DISCUSSÕES E RESULTADOS COMPUTACIONAIS

Nas próximas seções, são apresentados e discutidos os resultados numéricos provenientes das implementações dos algoritmos AGA, suas variações e o AGAH, previamente descritos.

### 5.1 ANÁLISE DAS MODIFICAÇÕES PROPOSTAS SOBRE O AG CLÁSSICO

Nesta seção, foram avaliadas as contribuições que cada modificação promoveu sobre o AG clássico para aumentar a qualidade nas soluções encontradas pelo método, bem como apresentado o ônus em termos de tempo de execução para as versões contendo cada uma das modificações.

Para isso foram propostas 4 versões de AGs modificados (AG Trechos, AG Rirnap, AG Padronizar e AG ACS), contendo as modificações isoladas e a versão clássica (AG Clássico). Testes foram realizados com as instâncias de até 400 cidades da TS-PLIB. Para cada uma das instâncias, gerou-se 20 populações iniciais fixas que foram utilizadas com as quatro versões modificadas e a versão clássica a fim de minimizar o efeito aleatório sobre as diferenças nos resultados finais.

Sobre os resultados executou-se o teste de hipóteses não paramétrico de *Wilcoxon Signed Rank* para duas amostras emparelhadas, comparando sempre, o desempenho entre uma das versões e o AG clássico.

Para realizar o teste, obtém-se a mediana da diferença entre as soluções apresentadas por ambas as metodologias e, avalia-se, se esta mediana é nula. A hipótese alternativa é de que a metodologia adaptada neste trabalho é superior a metodologia clássica e, neste caso, a mediana é maior que zero. O nível de significância ( $\alpha$ ) foi fixado em 0,05.

#### 5.1.1 COMPARAÇÃO ENTRE AG TRECHOS E O AG CLÁSSICO

Conforme discutido no Capítulo 3, esta modificação limita o comprimento dos trechos trocados pelos indivíduos durante a execução do operador de cruzamento. Na Figura 38 apresenta-se os resultados do teste Wilcoxon indicando que em 15% das instâncias houve diferença estatística significativa a favor da versão contendo o controle de trechos em comparação com a versão clássica. Em 85% das instâncias não

houve diferença e em nenhuma instância foram obtidos resultados significativamente melhores a favor da versão clássica de AG.

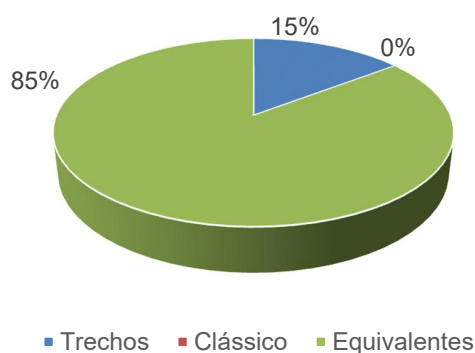


FIGURA 38: Resultado da Análise Estatística entre o AG Trechos e o AG Clássico

FONTE: Autor(2015)

O AG Trechos demonstrou requerer menos tempo por instância em relação ao AG Clássico, conforme pode ser observado na Figura 39. Para justificar a diferença de tempos de execução, argumenta-se que com o comprimento dos trechos sendo controlado é requerido menos tempo para possíveis reparações nos filhos em função dos trechos recebidos dos pais.

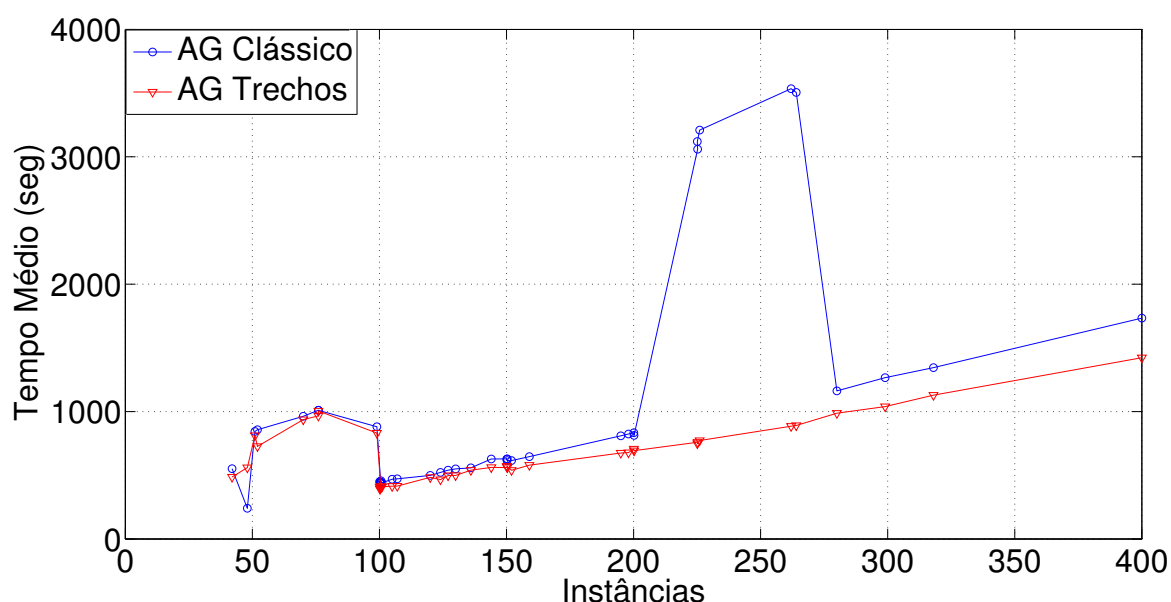


FIGURA 39: Tempo de Execução do AGATrechos e do AGClássico

FONTE: Autor(2015)

### 5.1.2 COMPARAÇÃO ENTRE AG ACS E O AG CLÁSSICO

Nesta subseção apresentam-se os resultados para a versão AG Acs que inclui a rotina Acs como parte dos critérios para determinar os indivíduos para cruzamento, conforme apresentado na subseção 3.5.5. No gráfico mostrado na Figura 40 pode-se observar que em 32% das instâncias o AG Acs forneceu resultados estatisticamente melhores que o AG Clássico, contra somente 4% a favor deste último e em 64% das instâncias as duas metodologias obtiveram desempenhos equivalentes.

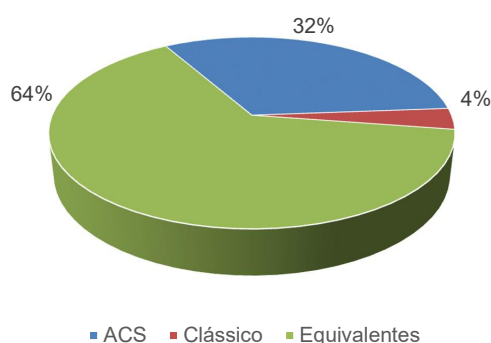


FIGURA 40: Resultado da Análise Estatística entre o AG Acs e o AG Clássico

FONTE: Autor(2015)

Esta modificação foi a responsável pela maior contribuição individual para com a diferença de desempenho entre o AGA e o AG Clássico. Contudo, o custo computacional é considerável por ser necessário calcular inúmeras vezes a similaridade entre dois indivíduos quaisquer ao longo das gerações. O acréscimo médio no tempo para

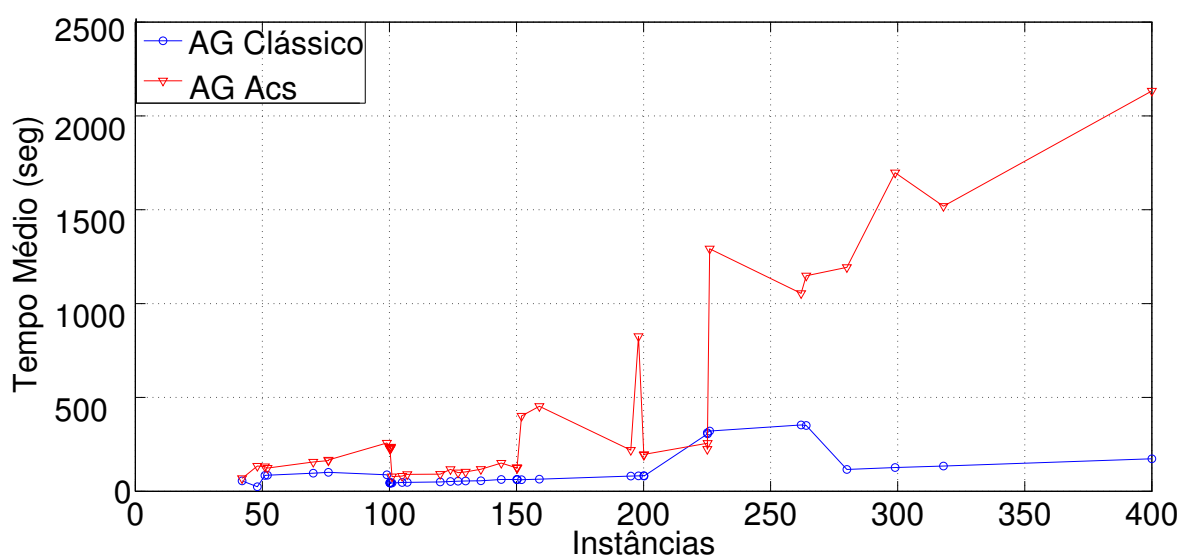


FIGURA 41: Tempo de Execução do AG Acs e do AG Clássico

FONTE: Autor(2015)

### 5.1.3 COMPARAÇÃO ENTRE AG RINAP E O AG CLÁSSICO

Nesta modificação, buscou-se aumentar a eficácia dos AGs inserindo indivíduos novos de forma aleatória na população a cada intervalo fixo de gerações. Novamente os resultados foram favoráveis à versão denominada AG Rinap. Esta implementação apresentou melhor desempenho comprovado estatisticamente em 17% das instâncias se comparado com o AG Clássico que foi melhor em 5% dos casos, enquanto que para as outras 78% das instâncias não foi possível admitir diferença entre as metodologias, conforme pode ser observado na Figura 42.

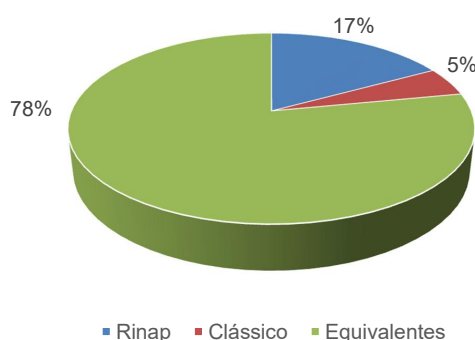


FIGURA 42: Resultado da Análise Estatística entre o AG Rinap e o AG Clássico

FONTE: Autor(2015)

Para o tempo de execução, o AG Rinap apresentou tempos médios de execução esporadicamente maiores, conforme observa-se na Figura 43 que apresenta a comparação entre os tempos médios de execução de ambas as versões.

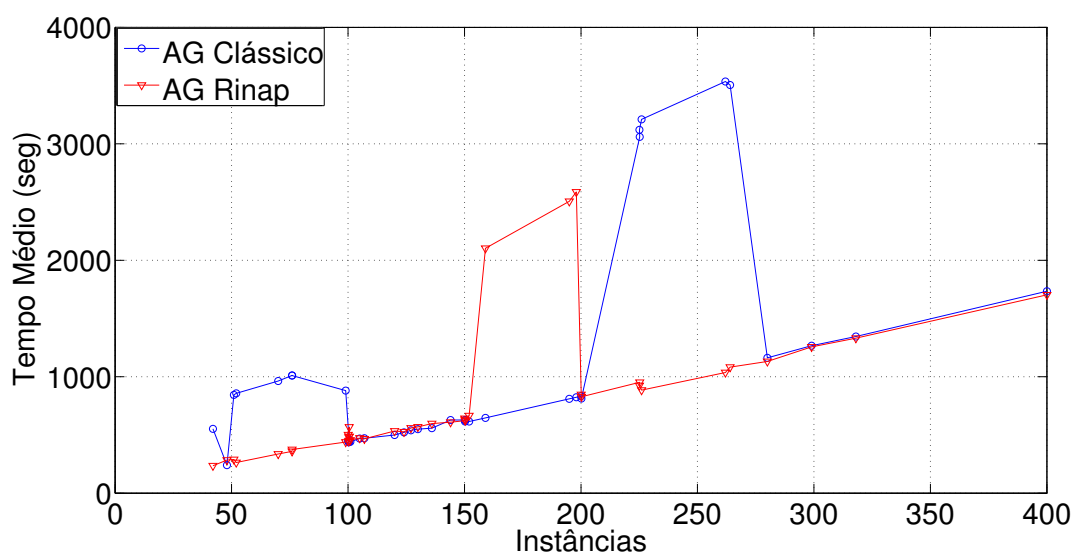


FIGURA 43: Tempo de Execução do AG Rinap e do AG Clássico

FONTE: Autor(2015)

#### 5.1.4 COMPARAÇÃO ENTRE AG PADRONIZAR E O AG CLÁSSICO

Entre as modificações propostas, esta foi a única versão que não superou o AG Clássico em porcentagem de instâncias com diferença estatística favorável à modificação. Conforme observa-se na Figura 44, em 27% das instâncias o AGClássico obteve resultados melhores contra somente 10% para o AGAPadronizar.

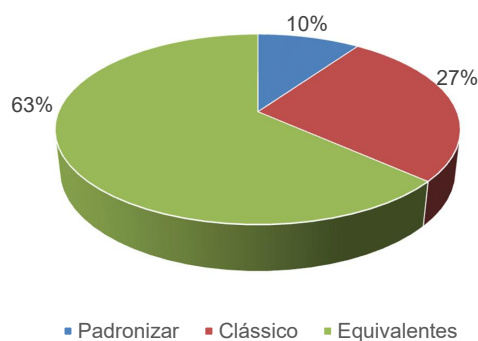


FIGURA 44: Resultado da Análise Estatística entre o AG Padronizar e o AG Clássico

FONTE: Autor(2015)

O tempo médio de execução do AG padronizar mostrou-se ligeiramente maior para instâncias pequenas, havendo um distanciamento nos tempos médios de execução à medida que instâncias maiores foram resolvidas, conforme pode ser observado na Figura 45. Esta diferença é natural uma vez que para realizar a padronização em grandes instâncias é requerido maior movimentação nas posições dos vetores que representam as soluções.

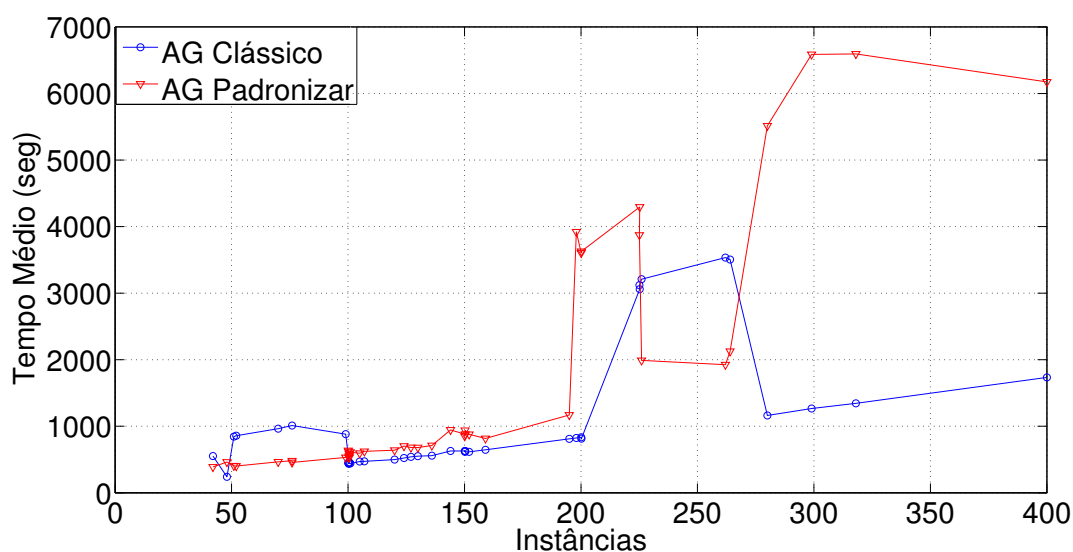


FIGURA 45: Tempo de Execução do AG Padronizar e o AG Clássico

FONTE: Autor(2015)

## 5.2 COMPARAÇÃO ENTRE AGA E O AG CLÁSSICO

Nesta seção constam os resultados para as instâncias com até 400 cidades, usando o AG *Steady-State* com elitismo, dois operadores clássicos de mutação e quatro operadores de cruzamento: PMX2, PMX3, OX4 e CNP4. Para o operador CNP4 o AG *Steady-State* com elitismo foi denominado AGA e recebeu as modificações no método de seleção de indivíduos adaptando o método do torneio com a rotina ACS, a inclusão da rotina RINAP que insere indivíduos, aleatoriamente, na população em intervalos fixos de gerações, o controle do comprimento dos trechos trocados entre os indivíduos durante a execução do operador de cruzamento e, por fim, a rotina de padronização da população.

Nestes testes, os AGs foram calibrados com os parâmetros da Tabela 7 da subseção 4.1.3. Os resultados podem ser observados na Tabela 11. Foram feitas 20 execuções para cada instância e apresentados o menor excesso (MenEx), o excesso médio (MedEx) e o maior excesso (MaxEx) em termos percentuais sobre a solução ótima.

Para melhor entendimento da Tabela 11 exemplificamos com o resultado 1,05 que consta na linha 2 e coluna 13 para o MedEx. Este número significa que em média, cada uma das 20 resoluções do AGA apresentou uma solução com aptidão que excede a ótima em cerca de 1,05% para a instância att48a.

TABELA 11: Comparação entre AG Clássico e AGA

Instância	Cid.	AG Clássico+PMX2			AG Clássico+PMX3			AG Clássico+OX4			AGA		
		Proximidade (%)			Proximidade (%)			Proximidade (%)			Proximidade (%)		
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx
dantzig42	42	5.43	13.40	27.40	7.31	10.68	15.57	0.00	0.42	2.40	0.00	0.00	0.00
att48a	48	3.16	14.84	27.36	3.39	9.81	17.28	0.19	1.96	4.21	0.19	1.05	3.49
eil51	51	5.40	14.87	24.44	5.05	11.16	14.17	0.70	2.11	4.43	0.00	2.00	4.02
Berlin52	52	17.51	26.03	37.07	10.78	16.72	21.77	0.00	1.85	5.43	0.00	1.70	5.50
st70	70	19.84	28.22	35.35	10.35	15.07	21.09	0.81	1.62	3.85	0.81	1.97	4.05
eil76a	76	8.97	17.26	21.57	12.74	18.94	25.74	2.60	10.10	32.93	1.03	3.07	4.95
pr76b	76	16.33	26.62	34.02	12.45	22.10	29.51	0.00	2.26	3.99	0.37	2.12	3.23
rat99	99	18.13	27.62	39.43	12.68	23.79	32.70	3.53	6.68	12.84	1.68	4.41	5.40
kroE100	100	18.03	28.89	38.32	14.29	22.23	32.20	0.94	2.24	4.21	0.49	2.18	6.56
rd100	100	25.11	29.98	36.47	16.99	23.97	29.76	1.00	2.77	5.35	0.01	2.29	7.16
kroC100	100	22.61	37.54	48.54	17.39	23.62	27.85	0.48	2.09	5.11	0.30	1.98	3.92
kroA100	100	25.88	32.68	42.46	4.17	19.90	32.14	0.43	1.51	3.27	0.10	0.82	2.64
kroD100	100	20.26	28.98	39.44	16.39	26.05	39.57	0.90	2.99	5.13	0.42	2.45	6.71
kroB100	100	17.05	30.87	41.11	18.22	23.51	31.38	1.14	2.84	5.17	1.07	1.59	2.11
eil101	101	18.52	24.67	31.64	10.66	18.18	23.76	2.42	5.82	9.67	0.23	3.71	5.01
lin105	105	23.65	34.88	53.31	11.67	22.47	32.09	0.26	2.19	3.93	0.00	1.09	3.76
pr107	107	19.70	28.24	32.80	6.46	15.58	22.44	0.63	0.99	1.26	0.00	0.53	1.04
gr120	120	24.21	29.10	33.91	16.72	22.83	30.03	4.73	7.14	9.91	1.79	3.96	7.53
pr124	124	20.90	32.58	47.05	14.22	22.56	27.19	0.78	2.68	6.95	0.00	2.29	4.92
bier127	127	18.32	30.47	42.02	14.89	24.99	33.24	3.65	8.50	14.24	1.62	5.56	10.24
ch130	130	31.86	38.95	44.25	19.25	27.62	34.97	5.25	7.60	11.04	1.18	3.81	7.00
pr136	136	17.24	25.38	34.20	13.07	25.75	34.14	2.42	5.52	9.22	1.45	4.08	6.93
pr144	144	56.38	65.23	74.50	20.75	36.07	45.70	0.20	2.19	4.48	0.11	3.05	6.64
kroB150	150	29.56	41.27	52.89	27.04	33.24	40.95	3.48	8.31	11.47	1.44	4.09	8.24
ch150	150	17.85	32.64	39.85	26.51	31.21	36.98	5.55	11.08	20.05	1.85	5.37	10.12
kroA150	150	30.04	38.03	52.53	27.26	34.10	40.42	5.94	9.12	13.04	0.85	3.80	6.44
pr152	152	20.60	36.85	45.24	15.30	26.05	35.99	2.31	4.31	6.92	1.11	2.70	5.08
u159	159	38.83	45.59	53.54	19.47	31.68	46.45	5.43	8.92	14.64	2.36	5.32	9.07
d198	198	26.86	33.27	43.38	15.51	24.06	31.33	6.44	9.59	12.16	2.21	4.67	7.49

continua para próxima página

continuação da página anterior													
Instância	Cid.	AG Clássico+PMX2			AG Clássico+PMX3			AG Clássico+OX4			AGA		
		Proximidade (%)			Proximidade (%)			Proximidade (%)			Proximidade (%)		
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx
kroA200	200	27.24	42.43	55.02	24.70	33.98	49.34	7.98	14.02	22.02	2.90	4.86	6.69
kroB200	200	27.56	40.36	47.57	23.24	34.01	42.86	7.65	13.92	18.89	2.89	6.02	8.68
tsp225	225	32.75	43.89	55.69	24.00	32.04	41.96	9.39	19.01	33.69	5.85	9.35	16.37
ts225	225	40.80	50.35	56.47	9.87	17.76	27.60	15.42	35.09	69.13	2.28	7.48	11.65
pr226	226	44.81	61.83	78.14	23.19	33.37	48.58	8.27	15.99	27.99	0.95	2.79	7.57
gil262	262	41.77	54.14	65.58	14.86	19.90	29.56	10.25	18.87	47.27	3.82	8.42	12.63
pr264	264	27.86	38.66	52.46	12.78	30.97	48.63	7.83	15.89	25.52	7.79	11.20	15.68
a280	280	28.11	32.76	36.68	19.26	26.79	33.08	18.79	31.65	71.36	5.03	9.76	12.87
pr299	299	29.65	34.42	40.49	19.45	28.73	33.63	15.85	24.20	35.28	5.19	9.00	17.34
lin318	318	29.94	38.63	47.57	11.23	15.53	20.23	13.61	25.35	43.51	5.53	9.27	14.87
rd400	400	53.30	56.89	65.54	12.04	14.33	16.18	25.89	30.26	36.96	11.63	16.98	23.36

FONTE: Autor(2015)

Para alguns problemas, o valor da coluna MaxEx do AGA é menor que o MenEx das outras 3 versões de AGs clássicos (como por exemplo em KroA200, a280, Rd400) apresentando forte indício de que para estas instâncias o AGA foi mais efetivo, porque todas as soluções por ele obtidas são melhores que qualquer solução de uma das outras versões.

Os resultados indicam que o conjunto de modificações foram capazes de agregar, para as instâncias testadas, melhor desempenho ao método. Para ratificar esta conclusão foi realizado o teste estatístico Mann-Whitney entre a versão de melhor desempenho do AG Clássico com o operador OX4 e o AGA com o operador CNP4.

O teste avalia, para cada instância, a hipótese nula de que as 20 soluções do AGA e do AG Clássico constituem amostras de uma mesma população, ou seja, representam metodologias equivalentes ao nível de significância de 5%, contra a hipótese alternativa de que as soluções do AGA são melhores. O resultado do teste é determinado pelo “valor p” presente na Tabela 12. Se o “valor p” for menor que 0,05 rejeita-se a hipótese nula e conclui-se que o AGA é mais eficiente.

TABELA 12: Teste Estatístico AG Clássico × AGA

Instância	Cidades	AG + OX4			AGA + CNP4			Mann Whitney		
		Proximidade (%)			Proximidade (%)			valor p	Amostra	Conclusão
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx			
att48a	48	0,192	1,962	4,207	0,192	1,048	3,486	0,030	20	AGA melhor
eil51	51	0,700	2,112	4,433	0,000	2,002	4,020	0,040	20	AGA melhor
<b>Berlin52</b>	<b>52</b>	<b>0,000</b>	<b>1,847</b>	<b>5,430</b>	<b>0,000</b>	<b>1,697</b>	<b>5,501</b>	<b>0,380</b>	20	<b>Equivalentes</b>
<b>st70</b>	<b>70</b>	<b>0,807</b>	<b>1,617</b>	<b>3,852</b>	<b>0,807</b>	<b>1,966</b>	<b>4,050</b>	<b>1,000</b>	20	<b>AGA pior</b>
eil76a	76	2,597	10,101	32,926	1,260	2,550	3,580	0,000	20	AGA melhor
pr76b	76	0,000	2,258	3,986	0,369	2,120	3,225	0,050	20	AGA melhor
rat99	99	3,527	6,677	12,840	1,679	4,407	5,398	0,004	20	AGA melhor
<b>kroE100</b>	<b>100</b>	<b>0,936</b>	<b>2,244</b>	<b>4,209</b>	<b>0,489</b>	<b>2,175</b>	<b>6,557</b>	<b>0,150</b>	20	<b>Equivalentes</b>
<b>rd100</b>	<b>100</b>	<b>0,996</b>	<b>2,772</b>	<b>5,346</b>	<b>0,011</b>	<b>2,286</b>	<b>7,159</b>	<b>0,120</b>	20	<b>Equivalentes</b>
kroC100	100	0,477	2,089	5,114	0,301	1,979	3,919	0,050	20	AGA melhor
kroA100	100	0,432	1,512	3,271	0,103	0,818	2,640	0,007	20	AGA melhor
<b>kroD100</b>	<b>100</b>	<b>0,898</b>	<b>2,989</b>	<b>5,133</b>	<b>0,423</b>	<b>2,453</b>	<b>6,707</b>	<b>0,150</b>	40	<b>Equivalentes</b>
kroB100	100	1,144	2,842	5,165	1,070	1,589	2,108	0,010	20	AGA melhor
eil101	101	2,420	5,820	9,673	0,230	3,706	5,007	0,003	20	AGA melhor

continua para próxima página



continuação da página anterior

Instância	Cidades	AG + OX4			AGA + CNP4			Mann Whitney		
		Proximidade (%)			Proximidade (%)			valor p	Amostra	Conclusão
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx			
lin105	105	0,257	2,188	3,925	0,000	1,089	3,759	0,002	20	AGA melhor
pr107	107	0,633	0,988	1,261	0,000	0,528	1,038	0,000	20	AGA melhor
gr120	120	4,725	7,143	9,914	1,791	3,964	7,527	0,002	20	AGA melhor
<b>pr124</b>	<b>124</b>	<b>0,782</b>	<b>2,682</b>	<b>6,954</b>	<b>0,000</b>	<b>2,286</b>	<b>4,922</b>	<b>0,080</b>	20	<b>Equivalentes</b>
bier127	127	3,651	8,496	14,240	1,618	5,560	10,236	0,012	20	AGA melhor
ch130	130	5,246	7,596	11,044	1,181	3,808	6,996	0,001	20	AGA melhor
pr136	136	2,418	5,521	9,217	1,445	4,080	6,925	0,050	20	AGA melhor
<b>pr144</b>	<b>144</b>	<b>0,199</b>	<b>2,191</b>	<b>4,477</b>	<b>0,106</b>	<b>3,046</b>	<b>6,642</b>	<b>1,000</b>	20	<b>AGA pior</b>
kroB150	150	3,484	8,305	11,466	1,442	4,091	8,243	0,000	20	AGA melhor
ch150	150	5,554	11,084	20,054	1,848	5,369	10,116	0,000	20	AGA melhor
kroA150	150	5,940	9,116	13,042	0,847	3,798	6,443	0,000	20	AGA melhor
pr152	152	2,307	4,313	6,918	1,106	2,695	5,076	0,007	20	AGA melhor
u159	159	5,430	8,923	14,644	2,363	5,322	9,065	0,012	20	AGA melhor
rat195	195	9,574	20,471	47,179	4,865	7,427	9,702	0,000	20	AGA melhor
d198	198	6,435	9,586	12,162	2,208	4,669	7,489	0,000	20	AGA melhor
kroA200	200	7,978	14,023	22,020	2,900	4,861	6,686	0,000	20	AGA melhor
<b>kroB200</b>	<b>200</b>	<b>7,646</b>	<b>13,919</b>	<b>18,893</b>	<b>2,892</b>	<b>6,024</b>	<b>8,680</b>	<b>0,500</b>	20	<b>Equivalentes</b>
tsp225	225	9,388	19,005	33,694	5,853	9,345	16,370	0,000	20	AGA melhor
ts225	225	15,420	35,085	69,134	2,281	7,480	11,647	0,000	20	AGA melhor
pr226	226	8,265	15,988	27,986	0,948	2,788	7,568	0,000	20	AGA melhor
gil262	262	10,248	18,871	47,274	3,822	8,417	12,625	0,001	20	AGA melhor
pr264	264	7,830	15,885	25,520	7,793	11,196	15,678	0,014	20	AGA melhor
a280	280	18,790	31,645	71,361	5,030	9,759	12,870	0,000	20	AGA melhor
pr299	299	15,854	24,204	35,281	5,185	9,002	17,342	0,000	20	AGA melhor
lin318	318	13,614	25,351	43,513	5,525	9,274	14,873	0,000	20	AGA melhor
rd400	400	25,888	30,258	36,958	11,634	16,982	23,359	0,001	20	AGA melhor

FORTE: Autor(2015)

Conforme observa-se na Tabela 12, em somente duas das 40 instâncias avaliadas, o AGA apresentou desempenho estatisticamente inferior aos AGs clássicos e em outras 6 instâncias não houve diferença significativa entre as metodologias, sendo que para as 32 instâncias restantes o AGA apresentou melhor desempenho. Desta forma, as modificações propostas são capazes de melhorar o desempenho do método. A Figura 46 apresenta os resultados da Tabela 12 em termos percentuais.

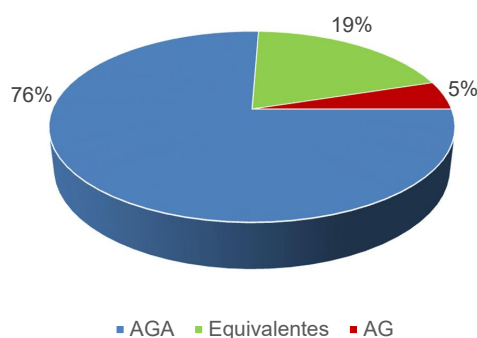


FIGURA 46: Resultado da Análise Estatística entre o AG e o AGA

FORTE: Autor(2015)

Em relação ao tempo de execução o AGA apresentou valores maiores em quase todos os casos, conforme pode-se observar na Figura 47.

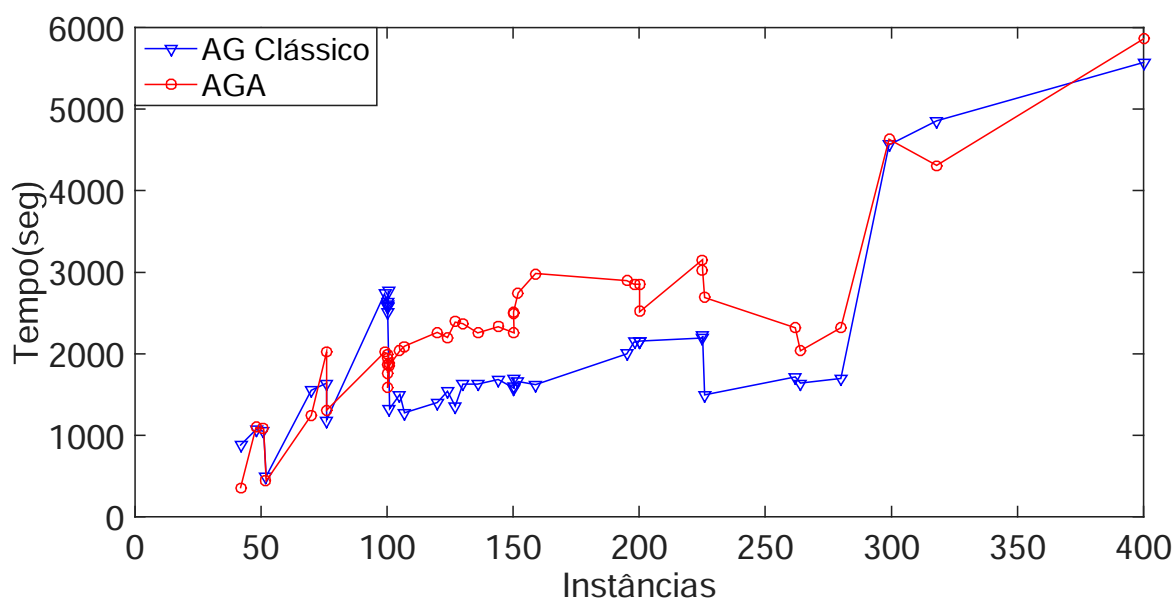


FIGURA 47: Tempo de execução do AG Clássico e do AGA

FONTE: Autor(2015)

### 5.3 MODIFICAÇÕES PROPOSTAS NA UTILIZAÇÃO DA HEURÍSTICA 3-OPT

O objetivo nesta seção é mostrar que o critério proposto para a ordem de escolha das cidades durante a execução da heurística 3-Opt a torna mais eficaz. Para isso, foram aplicadas duas metodologias: Na primeira aplicou-se a heurística 3-Opt com o critério (descrito no capítulo 3) sobre 100 soluções iniciais geradas aleatoriamente para cada uma das instâncias testadas; na segunda metodologia, repetiu-se o processo sem considerar o critério.

Os resultados podem ser observados na Tabela 13. Nas colunas MenEx, MedEx e MaxEx são apresentados, respectivamente, o menor excesso, o excesso médio e o maior excesso obtido sobre o ótimo em termos percentuais, após as 100 execuções para cada instância.

TABELA 13: 3-Opt Com Critério × 3-Opt Sem Critério

Instâncias	Cidades	Com Critério			Sem Critério			Mann Whitney	
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	valor p	Conclusão
st70	70	677,11	699,14	738,14	677,11	703,19	753,62	0,05	Melhor
eil76a	76	548,68	560,53	577,67	548,99	564,75	586,16	0,00	Melhor
pr76b	76	108234	112555	122320	108396	114053	131071	0,00	Melhor
gr96	96	54929	58111	62723	55231	59493	66690	0,00	Melhor
rat99	99	1223,6	1278,7	1330,1	1228,3	1296,7	1464,4	0,00	Melhor

continua para próxima página

continuação da página anterior

Instâncias	Cidades	Com Critério			Sem Critério			Mann Whitney	
		MenEx	MedEx	MaxEx	MenEx	MedEx	MaxEx	valor p	Conclusão
kroA100	100	22069	22574	24180	22116	22902	25444	0,00	Melhor
<b>kroB100</b>	<b>100</b>	<b>21285</b>	<b>21999</b>	<b>24698</b>	<b>21285</b>	<b>21984</b>	<b>24128</b>	<b>0,83</b>	<b>Equivalentes</b>
<b>kroC100</b>	<b>100</b>	<b>21346</b>	<b>22162</b>	<b>24352</b>	<b>21375</b>	<b>22291</b>	<b>24749</b>	<b>0,07</b>	<b>Equivalentes</b>
kroD100	100	21294	22025	23436	21294	22078	23957	0,00	Melhor
<b>kroE100</b>	<b>100</b>	<b>22069</b>	<b>22682</b>	<b>27202</b>	<b>22069</b>	<b>22691</b>	<b>24398</b>	<b>0,38</b>	<b>Equivalentes</b>
rd100	100	7911,3	8243,7	9207,4	7918,9	8364,8	9064,6	0,00	Melhor
eil101	101	647,02	660,7	682,09	646,5	663,28	675,21	0,00	Melhor
lin105	105	14383	14825	16048	14406	15015	16534	0,00	Melhor
pr107	107	44616	47359	61823	44684	48549	63473	0,03	Melhor
gr120	120	16257	16766	18920	16197	16818	18962	0,00	Melhor
pr124	124	59238	62058	69511	59246	64306	73226	0,00	Melhor
bier127	127	118727	122307	132152	119415	124137	131620	0,00	Melhor
ch130	130	6143	6303	6723	6125	6320	6658	0,00	Melhor
pr136	136	97729	101840	111657	98311	102967	109032	0,01	Melhor
gr137	137	70070	72399	77083	70124	73744	84358	0,00	Melhor
pr144	144	58621	64787	79145	61086	71617	89683	0,00	Melhor
ch150	150	6532	6755	7184	6532	6767	7274	0,00	Melhor
<b>kroA150</b>	<b>150</b>	<b>26572</b>	<b>27629</b>	<b>29972</b>	<b>26525</b>	<b>27620</b>	<b>30065</b>	<b>0,46</b>	<b>Equivalentes</b>
<b>kroB150</b>	<b>150</b>	<b>26163</b>	<b>26941</b>	<b>28777</b>	<b>26140</b>	<b>26936</b>	<b>28602</b>	<b>0,48</b>	<b>Equivalentes</b>
pr152	152	74528	79809	108224	75675	83381	106218	0,00	Melhor
u159	159	42235	43935	48881	42076	44753	50725	0,00	Melhor
si175	175	40730	41846	44263	41121	43153	47127	0,00	Melhor
rat195	195	2375,3	2441,3	2510,3	2377,6	2454,7	2635,4	0,00	Melhor
d198	198	15952	16881	19431	15902	17183	19459	0,00	Melhor
<b>kroA200</b>	<b>200</b>	<b>29417</b>	<b>30476</b>	<b>31878</b>	<b>29478</b>	<b>30498</b>	<b>32403</b>	<b>0,26</b>	<b>Equivalentes</b>
kroB200	200	29569	30677	33092	29470	30742	32929	0,01	Melhor
gr202	202	40519	41889	46065	40381	43149	49968	0,00	Melhor
tsp225	225	3939,9	4033,4	4238	3924,6	4072,4	4245,3	0,00	Melhor
<b>ts225</b>	<b>225</b>	<b>126812</b>	<b>134011</b>	<b>139882</b>	<b>126997</b>	<b>134782</b>	<b>144267</b>	<b>0,06</b>	<b>Equivalentes</b>
pr226	226	83750	105979	125998	93065	114745	136759	0,00	Melhor
gr229	229	136423	140096	160320	136113	146043	162398	0,00	Melhor
gil262	262	2406	2484,6	2569,4	2430,9	2490,8	2557,2	0,00	Melhor
pr264	264	50185	63098	77942	51377	64835	82236	0,00	Melhor
a280	280	2634,6	2756,7	2970,4	2658,4	2842	3131,4	0,00	Melhor
pr299	299	48812	51454	54802	49356	52249	58808	0,00	Melhor
lin318	318	42766	44064	47483	43352	45754	50638	0,00	Melhor
rd400	400	15633	15956	16388	15695	15976	16352	0,05	Melhor
<b>fl1417</b>	<b>417</b>	<b>14440</b>	<b>18662</b>	<b>24343</b>	<b>14145</b>	<b>18804</b>	<b>26161</b>	<b>0,36</b>	<b>Equivalentes</b>
gr431	431	175137	182817	214089	179374	194552	229053	0,00	Melhor
pr439	439	109571	121612	135494	111912	127063	144144	0,00	Melhor
pcb442	442	52143	53412	56457	52808	54911	59463	0,00	Melhor
d493	493	35835	37110	39490	36029	38542	42440	0,00	Melhor
att532	532	8880	9122,5	10078	8971	9285	10199	0,00	Melhor
<b>ali535</b>	<b>535</b>	<b>207358</b>	<b>227661</b>	<b>255217</b>	<b>210215</b>	<b>227044</b>	<b>251661</b>	<b>0,28</b>	<b>Equivalentes</b>
u574	574	38151	38890	41273	38643	41320	46388	0,00	Melhor
rat575	575	7008,6	7102,4	7188,4	7044,4	7252,7	7858,7	0,00	Melhor
d657	657	50123	51348	53642	52915	55691	61014	0,00	Melhor
gr666	666	309734	326220	352419	316443	348772	387673	0,00	Melhor
u724	724	43309	44145	45642	43398	44384	48325	0,00	Melhor
rat783	783	9157,4	9239,5	9405,1	9157	9429,3	9863,7	0,00	Melhor
dsj1000	1000	19223849	19957471	21647978	19272338	20064373	21754487	0,00	Melhor
pr1002	1002	276706	285974	302511	283351	305836	336483	0,00	Melhor
u1060	1060	240563	257443	277903	277891	316828	361918	0,00	Melhor
vm1084	1084	245800	249000	252890	244960	250550	254060	0,00	Melhor
pcb1173	1173	59459	60622	62489	60427	63911	71575	0,00	Melhor
d1291	1291	58605	64150	71553	61835	70307	77621	0,00	Melhor
rl1304	1304	272933	296466	318646	282929	312006	354049	0,00	Melhor
rl1323	1323	277811	286156	307529	279083	288598	302483	0,00	Melhor
nwr1379	1379	58960	60090	63836	59043	60371	63600	0,01	Melhor
<b>fl1400</b>	<b>1400</b>	<b>23909</b>	<b>31835</b>	<b>40684</b>	<b>22729</b>	<b>29614</b>	<b>42911</b>	<b>1,00</b>	<b>Pior</b>
fl1577	1577	31947	39654	45860	35529	43480	51819	0,00	Melhor
d1655	1655	72369	77461	86427	86642	94278	105296	0,00	Melhor
<b>vm1748</b>	<b>1748</b>	<b>348810</b>	<b>360380</b>	<b>389430</b>	<b>348320</b>	<b>355870</b>	<b>373800</b>	<b>1,00</b>	<b>Pior</b>
u1817	1817	63645	66541	70065	71525	79986	89174	0,00	Melhor
<b>rl1889</b>	<b>1889</b>	<b>329982</b>	<b>342883</b>	<b>366698</b>	<b>330608</b>	<b>341993</b>	<b>367136</b>	<b>1,00</b>	<b>Pior</b>
d2103	2103	99208	107723	115776	115008	127444	148663	0,00	Melhor
u2152	2152	70895	73114	76261	79912	86411	97849	0,00	Melhor
u2319	2319	240674	242557	244675	260497	273044	291316	0,00	Melhor
pr2392	2392	394379	401279	413820	400847	421509	453542	0,00	Melhor
pcb3038	3038	144113	145797	148165	144697	148318	156327	0,00	Melhor

FONTE: Autor(2015)

Com os dados da Tabela 13, observa-se que o critério de ordem de escolha das cidades para aplicação da heurística 3-Opt é capaz de melhorar significativamente as soluções finais, já que em sua grande maioria, as soluções apresentadas são melhores.

Por mais que houvesse evidências de melhorias na heurística 3-Opt após as modificações, aplicou-se um teste estatístico a fim de verificar a validade desta implementação. Assim como na seção anterior, optou-se pelo teste de Mann-Whitney, descrito em Geraldo (2012). A hipótese nula é de que não há diferença entre as metodologias “Com” e “Sem” o critério de ordem de escolha ao nível de significância de 5%, contra a hipótese alternativa de que os resultados são melhores ao incorporar o critério.

O resultado do teste foi apresentado ainda na Tabela 13, sendo que das 75 instâncias avaliadas, 9 apresentaram resultados sem diferença estatisticamente significativa para ambas as metodologias. Em apenas 3 instâncias, não usar o critério produziu melhores soluções, enquanto que para as outras 63 instâncias, a heurística 3-Opt com o critério apresentou melhor desempenho.

O acréscimo no tempo de execução da heurística quando se considera a ordem de escolha foi insignificante, conforme mostrado na Figura 48. Nesta figura apresenta-se, no eixo das ordenadas, o tempo computacional médio de cada instância para ambas as metodologias e as instâncias, no eixo das abcissas. É visível que ambas as metodologias tem aumento no tempo computacional médio equivalentes, à medida que são consideradas instâncias maiores.

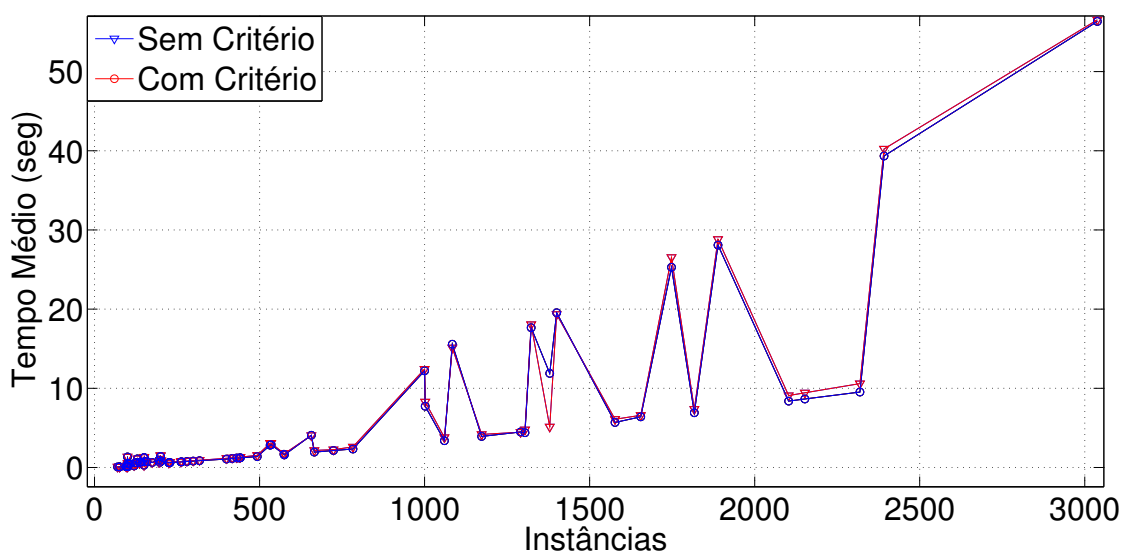


FIGURA 48: Tempo de execução a Heurística 3-Opt

FONTE: Autor(2015)

## 5.4 COMPARAÇÃO ENTRE O AGAH E O LK-H

Com o objetivo de validar o AGAH deste trabalho, os resultados foram comparados com os apresentados por Helsgaun (2000) por meio do algoritmo LK-H. O AGAH e o LK-H são metodologias diferentes, contudo o LK-H mostrou-se uma das melhores heurísticas da atualidade para o PCV. Se o AGAH for capaz de encontrar resultados equiparáveis com os obtidos por esta heurística, têm-se então bons argumentos para classificá-lo com uma ferramenta eficiente para o PCV.

Os resultados são apresentados em dois grupos: O primeiro é formado por problemas com até 400 cidades e o outro envolve problemas com mais de 400 cidades. Para o primeiro grupo, cada instância foi resolvida 100 vezes, e para o segundo 10 vezes. Esta diferença se deve ao excessivo tempo computacional que o AGAH requer para resolver instâncias maiores.

Na Tabela 14 são apresentados os resultados para instâncias com até 400 cidades, com a utilização da heurística AGAH e os resultados relatados em Helsgaun (2000) para este mesmo conjunto de instâncias. Dos 49 problemas comparados, em 43 deles as duas metodologias foram igualmente eficientes, encontrando a solução ótima nas 100 execuções. Em 5 problemas, o AGAH foi superior ao LK-H no sentido de encontrar mais vezes a solução ótima durante as 100 execuções. Em apenas 1 dos problemas o LK-H encontrou mais vezes a solução ótima.

Os 5 problemas em que o AGAH foi superior estão em negrito na Tabela 14. Nestes, verifica-se a efetividade do AGAH quando comparado ao LK-H. Conclui-se que o AGAH representa uma metodologia promissora na obtenção de soluções de boa qualidade para instâncias de tamanho menores, contudo, é visível que há uma grande diferença de tempo computacional. Portanto, ainda há muito a se evoluir nos AGs neste quesito.

TABELA 14: AGAH  $\times$  LK-H - Problemas com Até 400 Cidades

Nome	Algoritmo LK-H							AGA + CNP4 + MPH + MPHb + Mutação + 3-Opt						
	Sucessos	Proximidade (%)			Tempo (seg)			Sucessos	Proximidade (%)			Tempo (seg)		
		MenEx	MedEx	MaxEx	Pré	Média			MenEx	MedEx	MaxEx	Pré	Média	
gr24	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,26	8,10	
fri26	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	0,63	4,50	
bayg29	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,14	4,44	
bays29	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,21	10,49	
dantzig42	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,80	13,99	
swiss42	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,36	10,29	
att48a	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	1,20	42,11	
gr48b	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	1,31	15,44	
hk48	100	0,000	0,000	0,000	0,0	0,0		100	0,000	0,000	0,000	1,77	30,11	
eil51	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	3,26	44,45	
Berlin52	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	1,35	22,61	
Brasil58	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	2,36	18,28	
st70	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	0,78	68,77	
eil76a	100	0,000	0,000	0,000	0,1	0,0		100	0,000	0,000	0,000	1,29	224,36	
pr76b	100	0,000	0,000	0,000	0,1	0,1		100	0,000	0,000	0,000	2,16	37,04	
gr96	100	0,000	0,000	0,000	0,2	0,1		100	0,000	0,000	0,000	1,73	153,86	

continua para próxima página

continuação da página anterior													
Sucessos		Proximidade (%)			Tempo (seg)		Sucessos		Proximidade (%)			Tempo (seg)	
		MenEx	MedEx	MaxEx	Pré	Média			MenEx	MedEx	MaxEx	Pré	Média
rat99	100	0,000	0,000	0,000	2,0	0,0	100	0,000	0,000	0,000	1,71	243,75	
kroE100	99	0,000	0,002	0,172	0,2	0,2	100	0,000	0,000	0,000	1,58	299,57	
rd100	100	0,000	0,000	0,000	0,1	0,0	100	0,000	0,000	0,000	1,68	180,14	
kroC100	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	3,12	194,13	
kroA100	100	0,000	0,000	0,000	0,1	0,0	100	0,000	0,000	0,000	4,32	99,69	
kroD100	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	1,30	197,52	
kroB100	100	0,000	0,000	0,000	0,1	0,1	100	0,000	0,000	0,000	2,68	121,35	
eil101	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	1,62	777,20	
lin105	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	1,19	73,29	
pr107	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	2,48	945,62	
gr120	100	0,000	0,000	0,000	0,2	0,0	100	0,000	0,000	0,000	1,01	1313,52	
pr124	100	0,000	0,000	0,000	0,2	0,1	100	0,000	0,000	0,000	1,49	420,39	
bier127	100	0,000	0,000	0,000	0,3	0,0	100	0,000	0,000	0,000	3,54	1433,01	
ch130	100	0,000	0,000	0,000	0,2	0,1	100	0,000	0,000	0,000	1,40	929,30	
pr136	100	0,000	0,000	0,000	0,3	0,2	100	0,000	0,000	0,000	2,56	4016,94	
gr137	100	0,000	0,000	0,000	0,4	0,0	100	0,000	0,000	0,000	0,83	113,16	
pr144	100	0,000	0,000	0,000	0,3	0,1	100	0,000	0,000	0,000	1,42	236,73	
kroB150	55	0,000	0,003	0,008	0,3	0,8	100	0,000	0,000	0,000	2,12	905,56	
ch150	62	0,000	0,023	0,077	0,3	0,4	100	0,000	0,000	0,000	1,79	621,34	
kroA150	100	0,000	0,000	0,000	0,3	0,1	100	0,000	0,000	0,000	1,60	919,36	
pr152	100	0,000	0,000	0,000	0,5	0,1	100	0,000	0,000	0,000	1,57	425,03	
u159	100	0,000	0,000	0,000	0,3	0,0	100	0,000	0,000	0,000	9,45	212,89	
rat195	100	0,000	0,000	0,000	0,3	0,4	91	0,000	0,010	0,011	1,81	2726,20	
d198	100	0,000	0,000	0,000	0,4	1,0	100	0,000	0,000	0,000	1,85	5377,42	
kroA200	100	0,000	0,000	0,000	0,7	0,2	100	0,000	0,000	0,000	2,45	4597,81	
kroB200	100	0,000	0,000	0,000	0,4	0,1	100	0,000	0,000	0,000	3,20	20919,5	
tsp225	100	0,000	0,000	0,000	0,5	0,3	100	0,000	0,000	0,000	3,53	607,70	
ts225	100	0,000	0,000	0,000	0,5	0,1	100	0,000	0,000	0,000	7,31	4792,87	
pr226	100	0,000	0,000	0,000	0,6	0,1	100	0,000	0,000	0,000	2,29	3470,54	
gr229	12	0,000	0,009	0,010	0,7	1,7	100	0,000	0,000	0,000	2,36	11105,00	
gil262	100	0,000	0,000	0,000	0,6	0,4	100	0,000	0,000	0,000	3,09	1710,08	
a280	100	0,000	0,000	0,000	0,7	0,1	100	0,000	0,000	0,000	3,29	1627,26	
lin318	71	0,000	0,076	0,271	1,5	2,0	98	0,000	0,001	0,146	2,76	25078,64	

Fonte: Autor(2015)

Para problemas com mais de 400 cidades o AGAH mostrou-se menos eficiente que o LK-H. Os resultados são apresentados na Tabela 15. Nesta tabela, das 31 instâncias, em apenas 4 delas o AGAH obteve mais sucessos do que o LK-H. Na instância fl1577 é possível observar que o LK-H obteve a solução ótima 2 vezes, enquanto que com o AGAH foi mais efetivo pois obteve a solução ótima em todas as 10 vezes.

TABELA 15: AGAH × LK-H - Problemas com mais de 400 Cidades

Instância	Cidades	Algoritmo LKH								AGAH							
		Sucessos		Proximidade (%)			Tempo (seg)			Sucessos		Proximidade (%)			Tempo (seg)		
				MenEx	MedEx	MaxEx						MenEx	MedEx	MaxEx			
fl417	417	88	100	0.00	0.05	0.43	2.0	12.4	10	10	0.00	0.00	0.00	6.7	1366		
pr439	439	98	100	0.00	0.00	0.04	2.0	1.6	7	10	0.00	0.01	0.04	2.2	849		
pcb442	442	93	100	0.00	0.00	0.01	2.0	4.0	0	10	0.01	0.15	0.30	6.8	5715		
d493	493	100	100	0.00	0.00	0.00	2.5	3.2	4	10	0.00	0.10	0.25	4.4	5697		
att532	532	98	100	0.00	0.00	0.07	2.9	3.6	0	10	0.01	0.10	0.16	2.0	7555		
u574	574	91	100	0.00	0.01	0.08	3.3	3.2	6	10	0.00	0.05	0.17	4.4	5080		
rat575	575	77	100	0.00	0.00	0.03	3.2	8.2	2	10	0.00	0.07	0.26	4.8	8950		
p654	654	100	100	0.00	0.00	0.00	4.5	9.7	2	10	0.00	0.01	0.01	2.9	11662		
d657	657	100	100	0.00	0.00	0.00	5.1	3.3	0	10	0.04	0.12	0.25	4.9	11127		
gr666	666	30	100	0.00	0.03	0.04	4.8	18.8	9	10	0.00	0.00	0.07	2.7	28964		
u724	724	98	100	0.00	0.00	0.01	5.2	6.8	0	10	0.01	0.12	0.22	3.3	84674		
rat783	783	100	100	0.00	0.00	0.00	6.5	0.3	2	10	0.00	0.06	0.25	4.0	704243		
dsj1000	1000	7	10	0.00	0.04	0.12	12.3	55.1	0	10	0.01	0.09	0.22	5.5	664251		
pr1002	1002	10	10	0.00	0.00	0.00	15.6	3.4	0	10	0.01	0.49	1.11	1.1	11193		
u1060	1060	9	10	0.00	0.00	0.00	12.2	38.0	0	10	0.02	0.06	0.11	10.0	429610		
vm1084	1084	7	10	0.00	0.01	0.02	12.7	28.6	0	10	0.01	0.03	0.12	6.3	474884		
pcb1173	1173	8	10	0.00	0.00	0.01	15.7	14.6	0	10	0.03	0.26	0.47	10.3	470239		
nw1379	1379	3	10	0.00	0.01	0.01	23.3	69.3	0	10	0.26	0.50	0.72	15.1	84487		

continua para próxima página

continua para próxima página

continuação da página anterior																	
		Sucessos		Proximidade (%)			Tempo (seg)				Sucessos		Proximidade (%)			Tempo (seg)	
				MenEx	MedEx	MaxEx	Pré	Média					MenEx	MedEx	MaxEx	Pré	Média
<b>fl1400</b>	<b>1400</b>	<b>1</b>	<b>10</b>	<b>0.00</b>	<b>0.16</b>	<b>0.20</b>	<b>36.5</b>	<b>583.3</b>			<b>4</b>	<b>10</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	<b>12.6</b>	<b>44956</b>
u1432	1432	10	10	0.00	0.00	0.00	25.7	8.0			0	10	0.65	0.65	0.65	15.6	86747
<b>fl1577</b>	<b>1577</b>	<b>2</b>	<b>10</b>	<b>0.00</b>	<b>0.05</b>	<b>0.06</b>	<b>45.2</b>	<b>1097.5</b>			<b>10</b>	<b>10</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>11.8</b>	<b>42190</b>
d1655	1655	10	10	0.00	0.00	0.00	47.0	41.1			0	10	0.06	0.17	0.26	10.8	1209151
vm1748	1748	4	10	0.00	0.02	0.05	40.6	1016.1			0	10	0.06	0.38	0.76	10.8	1092002
u1817	1817	2	10	0.00	0.08	0.12	42.9	252.9			0	10	0.09	0.53	0.92	11.5	2918754
rl1889	1889	4	10	0.00	0.00	0.00	49.6	113.8			0	10	0.01	0.30	0.82	11.7	3048443
d2103	2103	0									0	10	0.02	0.18	0.34	23.2	3688104
u2152	2152	5	10	0.00	0.03	0.09	65.2	274.2			0	10	0.70	0.77	0.85	35.6	1060071
u2319	2319	10	10	0.00	0.00	0.00	86.2	2.6			0	10	1.29	1.29	1.29	39.0	1666735
pr2392	2392	10	10	0.00	0.00	0.00	84.6	54.5			0	10	0.35	0.57	0.69	25.4	740843
pcb3038	3038	9	10	0.00	0.00	0.00	139.8	323.7			0	10	0.74	0.79	0.85	47.6	1237020
fl3795	3795	4	10	0.00	0.07	0.19	220.9	5666.4			0	10	0.09	0.32	0.63	77.2	1251670

FONTE: Autor(2015)

## 6 CONCLUSÕES E RECOMENDAÇÕES

### 6.1 CONSIDERAÇÕES SOBRE O TRABALHO DESENVOLVIDO

Entre os objetivos deste trabalho, pode-se citar a busca por melhorar o desempenho de AGs, para PCVs, por meio de modificações em suas regras internas de funcionamento. Este objetivo foi atingido, uma vez que o desempenho do AGA para instâncias da TSPLIB foi superior ao desempenho de um AG clássico, conforme foi discutido no capítulo anterior.

Outro objetivo pretendido foi implementar um AG híbrido que apresentasse desempenho competitivo em relação ao LK-H, que é uma das melhores heurísticas presentes na literatura para resolver o PCV. Entre as potencialidades do LK-H estão sua rapidez de execução e a eficácia ao apresentar soluções de excelente qualidade (chegando a menos de 0,5% de excesso sobre a solução ótima para instâncias de grande porte e obtendo a própria solução ótima para instâncias menores). Para instâncias de grande porte o AGAH não se mostrou competitivo por requerer excessivo tempo de execução, mas para instâncias menores, com até 400 cidades, ele apresentou desempenho superior ao LK-H, conforme discutido nas análises do capítulo anterior. Desta forma, para instâncias menores, o objetivo foi atingido e a implementação apresentada neste trabalho é competitiva em relação ao LK-H.

Apesar de restrições impostas pelo excessivo tempo de execução para instâncias de grande porte, o AGAH apresentou desempenho satisfatório em termos de qualidade das soluções obtidas. Para algumas instâncias, contendo entre 1000 e 2000 cidades, a qualidades dos resultados apresentados pelo AGAH foram superiores ao LK-H o que corrobora sua regularidade na busca por boas soluções.

O novo método de seleção dos indivíduos para o cruzamento, proposto nesta tese, proporcionou maior eficácia para o AGA e o AGAH em relação ao método clássico do Torneio, conforme indicaram os testes da subseção 5.1.2. A ideia de um método de seleção que não considera somente o *fitness* dos indivíduos, mostrou-se produtiva e pesquisas futuras poderiam investir em novos critérios distintos e, possivelmente, mais aprimorados que o conceito de “similaridade” empregado.

O comprimento dos trechos durante a execução do operador de cruzamento também contribuiu para melhorar o desempenho do AGA e o AGAH, revelando-se mais uma modificação viável sobre o AG clássico, como se observa na subseção



5.1.1, por meio dos resultados dos testes estatísticos que compararam as versões com e sem esta modificação.

A modificação realizada sobre a heurística 3-Opt agregou benefícios estatisticamente comprovados, cujos experimentos se estenderam às instâncias da TSPLIB com até 3038 cidades. Os resultados demonstraram que, em somente 3 das 75 instâncias testadas não houve ganho de desempenho, portanto, esta modificação também representa numa opção viável em implementações futuras.

Dois dos três operadores genéticos de cruzamento propostos se destacaram por contribuir para acelerar a obtenção das melhores soluções. Ambos foram construídos considerando o mesmo conceito de “similaridade” do método de seleção. Estes operadores foram importantes para reduzir o tempo de execução do AGAH.

## 6.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Não foram propostas melhorias nos operadores genéticos de mutação, sendo aplicados somente dois operadores clássicos. Sugere-se para trabalhos futuros, inserir passos de 3-Opt, 4-Opt e 5-Opt como estratégias de mutação a fim de acelerar a obtenção de boas soluções no AG.

Uma sugestão para tornar o AG híbrido mais eficaz e reduzir seu tempo de execução é explorar a estrutura de árvore binária. Com isso, os operadores de cruzamento e mutação clássicos da literatura teriam que ser adaptados, contudo, a heurística 5-Opt poderia ser incorporada sem gerar acréscimos de tempo que inviabilizassem o AG.

Outra sugestão é implementar um AG com processamento em paralelo para aproveitar os recentes avanços no campo da informática através dos computadores com vários núcleos de processamento. Estas ações sem dúvida podem transformar os AGs numa excelente metodologia para resolução de grandes instâncias do PCV. Existem relatos de trabalhos bem sucedidos que investiram em processamento paralelo, tanto na CPU quanto na GPU, como é o caso de Honda, Nagata e Ono (2013) e Ghorpade e Kamalapur (2014).

Um recurso relatado em alguns trabalhos é a utilização de múltiplas populações evoluindo separadamente e, a uma baixa probabilidade, trocando indivíduos entre si. Em uma analogia ao mundo real, esta ação seria equivalente a existência de movimentos migratórios entre populações em regiões geograficamente dispersas.

Um desafio na implementação de uma heurística, diz respeito ao dimensionamento ideal dos parâmetros que controlam o modelo. Em nossa implementação 18

parâmetros influenciam consideravelmente, tanto no tempo de execução de uma instância qualquer, quanto na qualidade das soluções. Não foi realizado nenhum estudo sistemático para determinar o impacto da escolha dos valores dos parâmetros. Uma sugestão para trabalhos futuros é desenvolver uma metodologia cujos parâmetros sejam adaptativos, o que reduz sua influência no resultados finais.

Outro tópico que requer atenção em estudos futuros é o método de geração da população inicial. A abordagem proposta nesta tese é uma adaptação do vizinho mais próximo associado à geração totalmente aleatória. Percebeu-se empiricamente, que para algumas das 100 execuções de determinadas instâncias, o AGAH requeria um tempo extra de execução até obter a solução ótima, este fato pode estar relacionado a escolha da população inicial. Uma análise mais rigorosa do impacto do método de geração de soluções iniciais se faz necessário e pode agregar maior desempenho ao AG.

Finalmente recomenda-se implementar o método de seleção, a filosofia por trás da rotina que controla o comprimento dos trechos, a hibridização, o conceito de preservar partes comuns entre dois indivíduos que está por trás dos operadores MPH e MPHb e, principalmente, o conceito de “similaridade” em outros problemas para avaliar se o ganho de desempenho observado no AGA pode ser projetado aos AGs para problemas de natureza distinta do PCV.

## REFERÊNCIAS

APPLEGATE, D.; BIXBY, R.; CHVÁTAL, V. TSP Cuts Which Do Not Conform to the Template Paradigm. **Computational Combinatorial Optimization**, n. 1, p. 261–303, 2001. Disponível em: <[http://dx.doi.org/10.1007/3-540-45586-8\\_7](http://dx.doi.org/10.1007/3-540-45586-8_7)>.

APPLEGATE, D. *et al.* **Finding cuts in the TSP (A preliminary report)**. [S.l.]: Cite-seer, 1995.

APPLEGATE, D. *et al.* **Finding Tours in the TSP**. [s.n.], 1999. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:No+Title#0>>.

APPLEGATE, D. *et al.* Optimal tour of Sweden. **Last Updated June**, 2004.

APPLEGATE, D. *et al.* **On the solution of traveling salesman problems**. [S.l.]: Rheinische Friedrich-Wilhelms-Universität Bonn, 1998.

APPLEGATE, D. L. *et al.* The Traveling Salesman Problem: A Computational Study. In: **Operations Research**. [s.n.], 2007. p. 1–58. ISBN 9781400841103. Disponível em: <<http://press.princeton.edu/titles/8451.html>>.

BANZHAF, W. The 'molecular' traveling salesman. **Biological Cybernetics**, v. 64, n. 1, p. 7–14, 1990. ISSN 03401200.

BLAND, R. G.; SHALLCROSS, D. F. Large travelling salesman problems arising from experiments in X-ray crystallography: A preliminary report on computation. **Operations Research Letters**, v. 8, n. June, p. 125–128, 1989. ISSN 01676377.

BODIN, L. *et al.* Routing and scheduling of vehicles and crews. The state of the art. **Computers and Operations Research**, v. 10, n. 2, p. 63–211, 1983.

BORGES, S. K. **Resolução de Timetabling Utilizando Algoritmos Genéticos e Evolução Cooperativa**. Tese (Doutorado) — UFPR - Universidade Federal do Paraná, 2003.

BRUNETTA, L.; CONFORTI, M.; RINALDI, G. A branch-and-cut algorithm for the equicut problem. **Mathematical Programming**, v. 78, p. 243–263, 1997. ISSN 0025-5610.

CLERC, M. Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem. In: **New Optimization Techniques in Engineering**. [S.l.: s.n.], 2004. p. 219–239.

COOK, W. **Applications of the TSP**. 2014. The Traveling Salesman Problem p. Disponível em: <<http://www.math.uwaterloo.ca/tsp/index.html>>.

CROES, G. A. A method for solving traveling-salesman problems. **Operations Research**, v. 6, p. 794–812, 1958.

CROWDER, H.; PADBERG, M. W. Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality. **Management Science**, v. 26, n. 5, p. 495–509, 1980. ISSN 0025-1909.

CUNHA, A. G.; TAKAHASHI, R.; HENGGELE, C. **Manual de Computação Evolutiva e Metaheurística**. [S.l.]: Editora UFMG, 2013. 25–47 p. ISBN 9788542300468.

DAL, R. V. **Special Cases of the Traveling Salesman Problem**. [S.l.]: Wolters-Noordhoff, 1992.

DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a Large-Scale Traveling Salesman Problem. **Journal of the Operations Research Society of America**, p. 393–403, 1954.

DEEP, K.; MEBRAHTU, H. New Variations of Order Crossover for Travelling Salesman Problem. **International Journal of Combinatorial Optimization Problems and Informatics**, v. 2, n. 1, p. 2–13, 2011. ISSN 2007-1558. Disponível em: <<http://www.redalyc.org/resumen.oa?id=265219618002>>.

DUMITRESCU, I.; STUTZLE, T. A survey of methods that combine local search and exact algorithms. **FG Intellektik, FB Informatik, TU Darmstadt, Germany**, 2003.

FISCHETTI, M.; GONZALEZ, J. J. S.; TOTH, P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. **Operations Research**, v. 45, p. 378–394, 1997.

FOGEL, D. B. An evolutionary approach to the traveling salesman problem. **Biological Cybernetics**, v. 60, p. 139–144, 1988. ISSN 0340-1200. Disponível em: <<http://link.springer.com/10.1007/BF00202901>>.

GAMBOA, D. *et al.* An Experimental Evaluation of Ejection Chain Algorithms for the Traveling Salesman Problem. **Orp3 Meeting, Valencia. September 6-10, 2005**, p. 9 pp, 2006.

GERALDO, D. VERSÃO APRIMORADA DA METAHEURÍSTICA MULTIOBJECTIVE HARMONY SEARCH PARA PROBLEMAS COM VARIÁVEIS CONTÍNUAS. p. 2442–2453, 2012.

GHORPADE, S.; KAMALAPUR, S. Solution Level Parallelization of Local Search Metaheuristic Algorithm on GPU. **International Journal of Computer Science and Mobile Computing**, v. 3, n. 7, p. 268–274, 2014. Disponível em: <<http://www.ijcsmc.com/docs/papers/July2014/V3I7201464.pdf>>.

GLOVER, F. Tabu Search - Part I. **ORSA Journal on Computing**, v. 1, p. 190–206, 1989.

GOLDBERG, D. E. Alleles, loci, and the traveling salesman problem. In: **Proceedings of the first international conference on genetic algorithms and their applications**. [S.l.: s.n.], 1985. p. 154–159.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Addison Wesley, Reading, Massachusetts, 1989.

GRÖTSCHER, M. **On the symmetric travelling salesman problem: Solution of a 120-city problem**. [S.l.]: Springer, 1980.

GRÖTSCHER, M.; HOLLAND, O. Solution of large-scale symmetric travelling salesman problems. **Mathematical Programming**, Springer, v. 51, n. 1-3, p. 141–202, 1991.

GRÖTSCHER, M.; JÜNGER, M.; REINELT, G. Optimal control of plotting and drilling machines: A case study. **ZOR Methods and Models of Operations Research**, v. 35, p. 61–84, 1991. ISSN 03409422.

HELD, M.; KARP, R. The Traveling-Salesman Problem and Minimum Spanning Trees. **Operations Research**, v. 18, p. 1138–1162, 1970.

HELD, M.; KARP, R. M. A Dynamic Programming Approach to Sequencing Problems. **Journal of the Society for Industrial and Applied Mathematics**, v. 1, n. 3, p. 196–210, 1962.

HELGAUN, K. Effective implementation of the Lin-Kernighan traveling salesman heuristic. **European Journal of Operational Research**, v. 126, p. 106–130, 2000. ISSN 03772217.

HEPPNER, F.; GRENNANDER, U. A stochastic nonlinear model for coordinated bird flocks. **AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE**, 1990.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. [S.l.]: University Of Michigan Press, 1975.

HONDA, K.; NAGATA, Y.; ONO, I. A parallel genetic algorithm with edge assembly crossover for 100,000-city scale TSPs. In: **2013 IEEE Congress on Evolutionary Computation, CEC 2013**. [S.l.: s.n.], 2013. p. 1278–1285. ISBN 9781479904549.

ICHIHARA, J. d. A. **Algoritmos Genéticos**. 2013. Disponível em: <[www.eps.ufsc.br/teses98/ichihara/cap3.html](http://www.eps.ufsc.br/teses98/ichihara/cap3.html)>.

ISMKHAN, H.; ZAMANIFAR, K. Study of Some Recent Crossovers Effects on Speed and Accuracy of Genetic Algorithm , using Symmetric Travelling Salesman Problem. **International Journal of Computer Applications**, v. 80, n. 6, p. 6–11, 2013.

JOHNSON, D. S.; MCGEOCH, L. A. The traveling salesman problem: A case study in local optimization. **Local search in combinatorial optimization**, p. 215–310, 1997. Disponível em: <<http://142.103.6.5/hutter/previous-earg/EmpAlgReadingGroup/TSP-JohMcg97.pdf>>.

JÜNGER, M.; RINALDI, G.; REINELT, G. The Traveling Salesman Problem. **Handbooks in Operations Research and Management Science**, v. 7, p. 225–330, 1995.

KARP, H. D. R. M. On the computational complexity. **Networks**, v. 5, p. 45–68, 1975.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. **Neural Networks, 1995. Proceedings., IEEE International Conference on**, v. 4, p. 1942–1948 vol.4, 1995. ISSN 19353812.

KENNETH, A. d. J. **An analysis of the behavior of a class of genetic adaptive systems**. Tese (Doutorado) — University of MICHIGAN, 1975.

KIRKPATRICK, S.; GELATT, C. D. J.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v. 220, p. 671–680, 1983.

KNOX; GLOVER, F. Comparative Testing of Traveling Salesman Heuristics Derived from Tabu Search, Genetic Algorithms and Simulated Annealing. **Center for Applied Artificial Intelligence**, 1989.

KNOX, J. Tabu Search Performance on the Symmetric Traveling Salesman Problem. **Computers Ops. Res.**, v. 8, p. 867–876, 1994.

KURODA, M. *et al.* Development of a novel crossover of hybrid genetic algorithms for large-scale traveling salesman problems. **Artificial Life and Robotics**, v. 15, p. 547–550, 2010. ISSN 14335298.

LAND, A. H.; DOIG, A. G. An automatic method for solving discrete programming problems. **Econometrica**, v. 28, n. 3, p. 497–520, 1960. ISSN 0012-9682.

LAPORTE, G. The vehicle routing problem: An overview of exact and approximate algorithms. **European Journal of Operational Research**, v. 59, p. 345–358, 1992. ISSN 03772217.

LARRANAGA, P. *et al.* Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. **Artificial Intelligence Review**, v. 13, p. 129–170, 1999. Disponível em: <<http://dx.doi.org/10.1023/A:1006529012972>>.

LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal**, The, v. 44, p. 2245–2269, 1965.

LIN, S.; KERNIGHAN, B. W. **An Effective Heuristic Algorithm for the Traveling-Salesman Problem**. 1973. 498–516 p.

LIU, F.; ZENG, G. Study of genetic algorithm with reinforcement learning to solve the TSP. **Expert Systems with Applications**, Elsevier Ltd, v. 36, n. 3, p. 6995–7001, 2009. ISSN 09574174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2008.08.026>>.

LUCENA, A.; BEASLEY, J. A branch and cut algorithm for the Steiner problem in graphs. **Networks**, v. 31, p. 39–59, 1998. ISSN 10919856.

MACHADO, T. R.; LOPES, H. S. **A hybrid particle swarm optimization model for the traveling salesman problem**. [S.l.]: springer, 2005.

MATAI, R.; SINGH, S. P.; MITTAL, M. L. **Traveling Salesman Problem : An Overview of Applications, Formulations, and Solution Approaches**. [S.l.: s.n.], 2010. 1–24 p. ISBN 978-953-307-426-9.

MICHAEL, R. G.; DAVID, S. J. Computers and intractability: a guide to the theory of NP-completeness. **Freeman, San Francisco, CA**, 1979.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. [S.l.]: springer - Verlag Berlin Heidelberg, 1992. ISBN 3-540-60676-9.

MICHALEWICZ, Z.; FOGEL, D. B. Evolutionary Computation 2: Advance algorithms and operators. In: **IOP Institute of Physics**. [S.l.: s.n.], 2000. IOPC2000.

NETO, D. M. **Efficient cluster compensation for Lin-Kernighan heuristics**. Tese (Doutorado) — University of Toronto, 1999.

NGUYEN, H. D. *et al.* A New Three-Level Tree Data Structure for Representing TSP Tours in the Lin-Kernighan Heuristic. **IEICE Trans. Fundam. Electron. Commun. Comput. Sci.**, E90-A, p. 2187–2193, 2007. Disponível em: <<http://dx.doi.org/10.1093/ietfec/e90-a.10.2187>>.

NGUYEN, H. D. *et al.* Implementation of an effective hybrid GA for large-scale traveling salesman problems. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 37, n. 1, p. 92–99, 2007. ISSN 10834419.

PADBERG, M.; RINALDI, G. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. **Operations Research Letters**, v. 6, n. 1, p. 1–7, 1987. ISSN 0167-6377. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0167637787900022>>.

PADBERG, M.; RINALDI, G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. **SIAM review**, SIAM, v. 33, n. 1, p. 60–100, 1991.

PANG, W. E. *et al.* Modified Particle Swarm Optimization Based on Space Transformation for Solving Traveling Salesman Problem. **Nature**, n. V, p. 26–29, 2004.

PANWAR, P.; GUPTAH, S. Brief Survey of Soft Computing Techniques Used for Optimization of TSP. **The International Journal of Computer Science & Applications (TIJCSA)**, v. 2, n. 01, p. 54–59, 2013.

PLANTE, R. D.; LOWE, T. J.; CHANDRASEKARAN, R. The product matrix traveling salesman problem: an application and solution heuristic. **Operations Research**, v. 35, p. 772–783, 1987.

POTVIN, J. Y. Genetic algorithms for the traveling salesman problem. **Annals of Operations Research**, v. 63, p. 337–370, 1996. ISSN 0254-5330.

PRESTES, A. N. **Uma Análise Experimental de Abordagens Heurísticas Aplicadas ao Problema do Caixeiro Viajante**. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2006.

RATLIFF, H. D.; ROSENTHAL, a. S. Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. **Operations Research**, v. 31, n. 3, p. 507–521, 1983. ISSN 0030-364X.

RAVE, J. I. P.; ÁLVAREZ, G. P. J. Espaço literário relevante sobre el problema del vendedor viajero (TSP): contenido, clasificación, métodos y campos de inspiración. **Produção**, v. 23, p. 866–876, 2013. ISSN 01036513. Disponível em: <<http://dx.doi.org/10.1590/S0103-65132013005000003>>.

REINELT, G. **TSPLIB**. 1990. Disponível em: <<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>>.

REYES, H. R. **Solucion De Problemas De Optimizacion Mediante Algoritmos Geneticos Aplicando Computo De Alto Rendimiento**. Tese (Doutorado) — Instituto Politécnico Nacional, 2011.

REYNOLDS, C. W. Flocks, herds and schools: a distributed behavioral model. In: **Computer Graphics**. [S.l.: s.n.], 1987. v. 21, p. 24–34.

SOUZA, L. V. de. **Técnicas de Roteirização de Veículos Aplicadas ao Transporte Escolar**. Tese (Doutorado) — UFPR - Universidade Federal do Paraná, 1997. Disponível em: <<http://www.ppgmne.ufpr.br/dissertacoes/>>.

STEMMER, M. R. **Algoritmos Genéticos**. 2014.

STÜTZLE, T.; DORIGO, M. ACO Algorithms for the Traveling Salesman Problem. **Evolutionary Algorithms in Engineering and Computer Science**, p. 163–183, 1999.

TSAL, C.-w. *et al.* A High-Performance Genetic Algorithm : Using Traveling Salesman Problem as a Case. **Hindawi Publishing Corporation**, v. 2014, p. 1–15, 2014. Disponível em: <<http://dx.doi.org/10.1155/2014/178621>>.

WANG, K. P. *et al.* Particle swarm optimization for traveling salesman problem. In: **Machine Learning and Cybernetics, 2003 International Conference on**. [S.l.: s.n.], 2003. p. x1583–1585.

WANG, Y. The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem. **Computers & Industrial Engineering**, Elsevier Ltd, v. 70, p. 124–133, 2014. ISSN 03608352. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S036083521400031X>>.